



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**RANCANG BANGUN SISTEM *MONITORING* DAN
NOTIFIKASI *FALL DETECTION* MENGGUNAKAN
METODE *QUAD THRESHOLD* BERBASIS DUAL
SENSOR AKSELERASI DAN ORIENTASI
MULTIPOSISI**

BUTHORO KUNTO RAHARJO
05111440000068

Dosen Pembimbing
Waskitho Wlbisono, S.Kom., M.Eng., Ph.D.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KI141502

**RANCANG BANGUN SISTEM *MONITORING* DAN
NOTIFIKASI *FALL DETECTION* MENGGUNAKAN
METODE *QUAD THRESHOLD* BERBASIS DUAL
SENSOR AKSELERASI DAN ORIENTASI
MULTIPOSISI**

BUTHORO KUNTO RAHARJO
05111440000068

Dosen Pembimbing I
Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

***DESIGN AND DEVELOPMENT OF FALL
DETECTION MONITORING AND NOTIFICATION
SYSTEM BASED ON QUAD THRESHOLD METHOD
FOR DUAL ACCELERATION AND ORIENTATION
MULTIPOSITION SENSOR***

BUTHORO KUNTO RAHARJO
05111440000068

Supervisor I
Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM MONITORING DAN NOTIFIKASI FALL DETECTION MENGGUNAKAN METODE QUAD THRESHOLD BERBASIS DUAL SENSOR AKSELERASI DAN ORIENTASI MULTIPOSISI

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi berbasis Jaringan
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

BUTHORO KUNTO RAHARJO

NRP: 0511144000068

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Waskitho Wibisono, S.Kom., M.Eng., Ph.D.
NIP. 197410222000031001



(Pembimbing 1)

**SURABAYA
JULI, 2018**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SISTEM MONITORING DAN NOTIFIKASI FALL DETECTION MENGGUNAKAN METODE QUAD THRESHOLD BERBASIS DUAL SENSOR AKSELERASI DAN ORIENTASI MULTIPOSISI

Nama Mahasiswa : Buthoro Kunto Raharjo
NRP : 05111440000068
Departemen : Informatika, FTIK ITS
Dosen Pembimbing 1 : Waskitho Wibisono, S.Kom, M.Eng.,
Ph.D.
Dosen Pembimbing 2 : -

Abstrak

Jatuh merupakan kondisi dimana seseorang secara tidak sengaja tersungkur ke bawah. Jatuh sering kali merupakan penyebab seseorang mengalami luka dan bahkan tidak sedikit yang berujung kematian. Pada orang yang sudah berada pada usia lanjut, risiko yang diakibatkan dari jatuh akan lebih besar dibanding dengan orang yang masih berusia muda. Oleh karena itu perlu adanya sebuah sistem monitoring yang berguna untuk membantu seseorang yang dalam keadaan setelah jatuh yang dimungkinkan pada saat itu mengalami keadaan tidak sadar.

Berbagai penelitian telah dilakukan untuk menanggulangi masalah tersebut. Tetapi peningkatan sensitivitas, spesifitas dan akurasi dirasa perlu karena pada saat ini, penelitian yang mengangkat masalah ini masih perlu banyak pengembangan lebih lanjut.

Pada tugas akhir ini, sistem dibuat menggunakan masing – masing dua sensor accelerometer dan gyroscope yang diintegrasikan pada microcontroller arduino yang kemudian ditempatkan pada pakaian pengguna. Microcontroller arduino bertugas mengatur pengambilan data oleh sensor, pengolahan data dan penentuan keputusan gerakan jatuh pengguna. Data hasil pengolahan dari microcontroller arduino dikirimkan ke perangkat android melalui Bluetooth. Dari perangkat android ditentukan

posisi pengguna kemudian notifikasi akan dikirim ke keluarga atau kerabat pengguna jika pengguna positif terdeteksi jatuh.

Kata kunci: Accelerometer, Arduino, Fall Detection, Gyroscope, Jatuh, Multi Sensor.

**DESIGN AND DEVELOPMENT OF FALL DETECTION
MONITORING AND NOTIFICATION SYSTEM BASED ON
QUAD THRESHOLD METHOD FOR DUAL
ACCELERATION AND ORIENTATION MULTIPOSITION
SENSOR**

Student Name : Buthoro Kunto Raharjo
Registration Number : 05111440000068
Department : Informatics, FTIK ITS
First Supervisor : Waskitho Wibisono, S.Kom., M.Kom.,
Ph.D.
Second Supervisor : -

Abstract

Falling is a condition where someone accidentally drops down. Falling is often the cause of a person suffering injuries and not even a few that lead to death. In people who are already elderly, the risk resulting from a fall will be greater than that of a young person. Therefore it is necessary to have a monitoring system useful to assist a person who is in a state after falling which is possible at that time to experience an unconscious state.

Various studies have been done to tackle the problem. But increased sensitivity, specificity and accuracy are necessary because at present, research raising this issue still needs much further development.

In this final project, the system is made using two accelerometer and gyroscope sensors that are integrated into arduino microcontroller which is then placed on the user's clothing. The arduino microcontroller is responsible for organizing data retrieval by sensors, data processing and user falling decision decisions. Processed data from arduino microcontroller is sent to android device via Bluetooth. From the android device specified user position then notification will be sent to the family or relatives of the user if positive user is detected to fall.

Keywords : Accelerometer, Arduino, Fall, Fall Detection, Gyroscope, Multi Sensor.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah Subhana Wa Ta'ala yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Rancang Bangun Sistem Monitoring Dan Notifikasi Fall Detection Menggunakan Metode Quad Threshold Berbasis Dual Sensor Akselerasi Dan Orientasi Multiposisi”**.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam penelitian deteksi jatuh pada seseorang lebih lanjut. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Departemen Informatika ITS.

Dalam perancangan, pengerjaan dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak, sehingga penulis ingin mengucapkan terima kasih kepada:

1. Allah Subhana wa Ta'ala dan Nabi Muhammad Shallallahu 'alaihi wa Salam
2. Bapak Triman dan Ibu Sumarmi selaku orang tua penulis yang selalu memberikan dukungan doa, moral, dan material yang tak terhingga kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D. selaku pembimbing I yang telah membimbing dan memberikan motivasi, nasehat dan bimbingan dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknik Informatika ITS.
5. Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc. selaku koordinator Tugas Akhir dan segenap dosen Departemen Informatika yang telah memberikan ilmunya.

6. Febryana Nur Safitri, S.Ked., sahabat penulis yang telah membantu, membagi ilmu dan memberi motivasi kepada penulis.
7. Teman-teman Ikatan Keluarga Mahasiswa Sukoharjo di Surabaya (Ikemas Surabaya).
8. Cahya, Muhsin, Cepot, Anggoro, Fahny, Menco, Ari, Bagas, Jul, Bintang dan teman-teman seperjuangan yang selalu menemani dan membantu penulis pada saat mengerjakan Tugas Akhir.
9. Teman-teman angkatan 2014, yang sudah mendukung penulis selama perkuliahan.
10. Sahabat penulis yang tidak dapat disebutkan satu per satu yang selalu membantu, menghibur, menjadi tempat bertukar ilmu dan berjuang bersama-sama penulis.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER.....	xxiii
1 BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan Tugas Akhir	3
1.5 Manfaat Tugas Akhir	4
1.6 Metodologi	4
1.1.1 Penyusunan Proposal.....	4
1.1.2 Studi Literatur	4
1.1.3 Perancangan dan Desain Sistem.....	5
1.1.4 Implementasi Sistem.....	5
1.1.5 Pengujian dan Evaluasi.....	5
1.1.6 Penyusunan Buku	5
1.7 Sistematika Laporan.....	6
2 BAB II TINJAUAN PUSTAKA.....	7
2.1 <i>Microcontroller Arduino</i>	7
2.2 Arduino Software (IDE).....	8
2.3 Sensor <i>Accelerometer</i>	8
2.4 Sensor <i>Gyroscope</i>	9
2.5 Modul GY – 52 MPU – 6050	9
2.6 <i>Bluetooth</i>	11
2.7 Android.....	11
2.8 Android Studio.....	12
3 BAB III PERANCANGAN DAN DESAIN SISTEM	15

3.1	Deskripsi Umum Sistem.....	15
3.2	Arsitektur Umum Sistem.....	17
3.3	Perancangan Integrasi <i>Microcontroller Arduino</i> dengan Perangkat Sensor	19
3.4	Perancangan Diagram Alir	20
3.4.1	Inisialisasi <i>Microcontroller Arduino</i>	20
3.4.2	Pembacaan Nilai <i>Accelerometer</i>	20
3.4.3	Pembacaan Nilai <i>Gyroscope</i>	21
3.4.4	Algoritma <i>Fall Detection</i> Menggunakan Metode <i>Quad Threshold</i>	22
3.4.5	Klasifikasi Posisi Berdiri	23
3.4.6	Penentuan Jatuh.....	23
3.4.7	Pengiriman Notifikasi.....	24
3.5	Perancangan Antarmuka Sistem	28
3.5.1	Rancangan Antarmuka Awal Aplikasi	29
3.5.2	Rancangan Antarmuka Tambah Kontak.....	30
3.5.3	Rancangan Antarmuka Daftar Perangkat <i>Bluetooth</i>	30
3.5.4	Rancangan Antarmuka Proses Deteksi Jatuh.....	31
4	BAB IV IMPLEMENTASI	33
4.1	Lingkungan Implementasi	33
4.2	Implementasi Perangkat Keras	34
4.3	Implementasi pada <i>Microcontroller Arduino</i>	36
4.3.1	Implementasi Inisialisasi <i>Microcontroller Arduino</i>	36
4.3.2	Implementasi Pembacaan Sensor	37
4.3.3	Implementasi Pengambilan Data <i>Training</i>	38
4.3.4	Implementasi Algoritma <i>Fall Detection</i> Menggunakan Metode <i>Quad Threshold</i>	51
4.4	Implementasi Aplikasi Android <i>Fall Detection</i>	54
4.4.1	Implementasi Perizinan Menggunakan <i>Bluetooth</i> , Lokasi, dan Mengirimkan SMS pada Perangkat Android	54
4.4.2	Implementasi Fitur Menambah, Mengubah dan Menghapus Kontak.....	55
4.4.3	Implementasi Fitur Mencari Perangkat <i>Bluetooth</i>	57

4.4.4 Implementasi Koneksi Perangkat Android dengan Perangkat <i>Bluetooth</i>	58
4.4.5 Implementasi Proses Deteksi Jatuh dan Pengiriman Notifikasi Sms	59
5 BAB V HASIL UJI COBA DAN EVALUASI	61
5.1 Lingkungan Uji Coba	61
5.2 Data Uji Coba	62
5.3 Skenario Uji Coba.....	67
5.3.1 Uji Coba Fungsionalitas.....	67
5.3.2 Uji Coba Performa Sistem	80
5.3.3 Analisis Hasil Uji Coba	85
6 BAB VI KESIMPULAN DAN SARAN	87
6.1 Kesimpulan.....	87
6.2 Saran.....	88
DAFTAR PUSTAKA	89
LAMPIRAN	93
BIODATA PENULIS.....	127

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Microcontroller Arduino [9].....	7
Gambar 2.2 Arduino Software.....	8
Gambar 2.3 Modul GY - 52	10
Gambar 2.4 Android Studio	13
Gambar 3.1 Posisi penempatan sensor dan <i>microcontroller arduino</i>	18
Gambar 3.2 Arsitektur Umum Sistem.....	19
Gambar 3.3 Rangkaian integrasi <i>microcontroller arduino</i> dengan perangkat sensor	19
Gambar 3.4 Diagram alir inisialisasi <i>microcontroller arduino</i> ...	20
Gambar 3.5 Diagram alir pembacaan nilai <i>accelerometer</i>	21
Gambar 3.6 Diagram alir pembacaan nilai <i>gyroscope</i>	22
Gambar 3.7 Diagram alir algoritma <i>fall detetction</i> menggunakan metode <i>quad threshold</i>	25
Gambar 3.8 Diagram alir klasifikasi posisi berdiri.....	26
Gambar 3.9 Diagram alir penentuan jatuh.....	27
Gambar 3.10 Diagram alir pengiriman notifikasi	28
Gambar 3.11 Rancangan antarmuka awal aplikasi	29
Gambar 3.12 Rancangan antarmuka tambah kontak.....	30
Gambar 3.13 Rancangan antarmuka daftar perangkat <i>bluetooth</i>	31
Gambar 3.14 Rancangan antarmuka proses deteksi jatuh	32
Gambar 4.1 Rangkaian perangkat sensor	35
Gambar 4.2 Grafik data <i>training accelerometer</i> pada aktivitas berdiri dan berjalan	39
Gambar 4.3 Grafik data <i>training gyroscope</i> pada aktivitas berdiri dan berjalan	40
Gambar 4.4 Grafik resultan kecepatan sudut ($\Delta\omega$)	41
Gambar 4.5 Grafik data <i>training accelerometer</i> pada aktivitas transisi dari posisi berdiri ke posisi duduk	42
Gambar 4.6 Grafik data <i>training gyroscope</i> pada aktivitas transisi dari posisi berdiri ke posisi duduk	42
Gambar 4.7 Grafik resultan kecepatan sudut ($\Delta\omega$)	43
Gambar 4.8 Grafik data <i>training accelerometer</i> pada aktivitas merebahkan badan	44

Gambar 4.9 Grafik data <i>training gyroscope</i> pada aktivitas merebahkan badan	45
Gambar 4.10 Grafik selisih resultan percepatan linear (Δa).....	45
Gambar 4.11 Grafik selisih resultan kecepatan sudut ($\Delta \omega$).....	46
Gambar 4.12 Grafik perubahan sudut (θ)	47
Gambar 4.13 Grafik data <i>training accelerometer</i> pada gerakan jatuh	48
Gambar 4.14 Grafik data <i>training gyroscope</i> pada gerakan jatuh.....	48
Gambar 4.15 Grafik selisih resultan percepatan linear (Δa)	49
Gambar 4.16 Grafik selisih resultan kecepatan sudut ($\Delta \omega$)	50
Gambar 4.17 Grafik perubahan sudut (θ)	50
Gambar 5.1 Grafik data <i>accelerometer</i> uji coba berjalan.....	62
Gambar 5.2 Grafik data <i>gyroscope</i> uji coba berjalan.....	63
Gambar 5.3 Grafik data <i>accelerometer</i> uji coba transisi.....	64
Gambar 5.4 Grafik data <i>gyroscope</i> uji coba transisi.....	64
Gambar 5.5 Grafik data <i>accelerometer</i> uji coba merebahkan badan	65
Gambar 5.6 Grafik data <i>gyroscope</i> uji coba merebahkan badan	65
Gambar 5.7 Grafik data <i>accelerometer</i> uji coba jatuh	66
Gambar 5.8 Grafik data <i>gyroscope</i> uji coba jatuh	67
Gambar 5.9 Halaman awal aplikasi	68
Gambar 5.10 Halaman awal aplikasi ketika menampilkan daftar kontak.....	69
Gambar 5.11 Halaman tambah kontak.....	69
Gambar 5.12 Halaman awal ketika data kontak berhasil disimpan	70
Gambar 5.13 Halaman lihat kontak	71
Gambar 5.14 Halaman untuk mengubah kontak.....	71
Gambar 5.15 Halaman awal ketika berhasil mengubah kontak..	72
Gambar 5.16 Halaman lihat kontak	73
Gambar 5.17 Halaman awal ketika pengguna berhasil menghapus kontak.....	73
Gambar 5.18 Halaman pencarian perangkat <i>bluetooth</i> ketika meminta aktivasi <i>bluetooth</i>	74

Gambar 5.19 Halaman pencarian perangkat <i>bluetooth</i> ketika melakukan pencarian perangkat.....	75
Gambar 5.20 Halaman pencarian perangkat <i>bluetooth</i> ketika menampilkan daftar perangkat hasil pencarian.....	75
Gambar 5.21 Halaman <i>fall detection</i> ketika perangkat android sudah terkoneksi dengan perangkat <i>bluetooth</i> dan sedang melakukan proses deteksi jatuh	77
Gambar 5.22 Halaman <i>fall detection</i> ketika perangkat android tidak terkoneksi dengan perangkat <i>bluetooth</i>	77
Gambar 5.23 Halaman <i>fall detection</i> ketika pengguna terdeteksi jatuh dan aplikasi mengirimkan notifikasi berupa sms	78
Gambar 5.24 Sms terkirim yang ditampilkan melalui perangkat android pengguna.....	79
Gambar 5.25 Sms diterima yang ditampilkan melalui perangkat android keluarga / kerabat pengguna	79
Gambar 5.26 Hasil tautan lokasi jatuh pengguna yang dibuka melalui perangkat android keluarga / kerabat pengguna.....	80
Gambar 5.27 Posisi pemasangan perangkat sensor pada pengguna	81
Gambar 5.28 Analisis hasil uji coba	86

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Lingkungan implementasi.....	33
Tabel 5.1 Lingkungan pengujian	61
Tabel 5.2 Hasil uji coba jatuh pada skenario uji coba 1	82
Tabel 5.3 Hasil uji coba memperagakan aktivitas sehari - hari pada skenario uji coba 2	84
Tabel 6.1 Data <i>training</i> aktivitas berjalan pada sensor 2	93
Tabel 6.2 Data <i>training</i> gerakan transisi pada sensor 2.....	97
Tabel 6.3 Data <i>training</i> gerakan merebahkan badan pada sensor 1	101
Tabel 6.4 Data <i>training</i> gerakan jatuh pada sensor 1	103
Tabel 6.5 Contoh data uji coba aktivitas berjalan pada sensor 1	105
Tabel 6.6 Contoh data uji coba aktivitas berjalan pada sensor 2	111
Tabel 6.7 Contoh data uji coba gerakan transisi pada sensor 2	118
Tabel 6.8 Contoh data uji coba gerakan merebahkan badan pada sensor 1.....	121
Tabel 6.9 Contoh data uji coba gerakan merebahkan badan pada sensor 2.....	123
Tabel 6.10 Contoh data uji coba gerakan jatuh pada sensor 1..	125

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Inisialisasi pada <i>microcontroller arduino</i>	37
Kode Sumber 4.2 Implementasi pembacaan sensor.....	38
Kode Sumber 4.3 Implementasi algoritma <i>fall detection</i> untuk sensor 1.....	53
Kode Sumber 4.4 Implementasi algoritma <i>fall detection</i> untuk sensor 2.....	53
Kode Sumber 4.5 Melakukan pengecekan apakah perangkat android sudah memiliki <i>bluetooth</i>	54
Kode Sumber 4.6 Melakukan pengecekan apakah perangkat android <i>support</i> dengan <i>bluetooth low energy</i>	55
Kode Sumber 4.7 Izin untuk menggunakan dan mengaktifkan <i>bluetooth</i> jika belum diaktifkan	55
Kode Sumber 4.8 Izin untuk menggunakan lokasi perangkat android	55
Kode Sumber 4.9 izin untuk menggunakan fitur sms pada perangkat android	55
Kode Sumber 4.10 Menambahkan data kontak ke <i>database</i> berdasarkan <i>inputan</i> pengguna	56
Kode Sumber 4.11 Mengubah data kontak	57
Kode Sumber 4.12 Menghapus kontak dari <i>database</i>	57
Kode Sumber 4.13 Fungsi mencari dan mengatur <i>delay</i> pencarian perangkat <i>bluetooth</i>	58
Kode Sumber 4.14 Fungsi menyimpan hasil pencarian perangkat <i>bluetooth</i> ke dalam <i>list</i>	58
Kode Sumber 4.15 Melakukan inisialisasi <i>bluetooth service</i> dan memanggil fungsi <i>connect</i> pada <i>class BluetoothLeService</i> untuk melakukan koneksi perangkat android dengan perangkat <i>bluetooth</i>	59
Kode Sumber 4.16 Fungsi <i>connect</i> pada <i>class BluetoothLeDevice</i>	59
Kode Sumber 4.17 Fungsi untuk mendapatkan data secara terus menerus dari perangkat <i>bluetooth</i>	60
Kode Sumber 4.18 Fungsi <i>BroadcastReceiver</i> untuk menampilkan data pada halaman <i>fall detection</i> dan mengirimkan notifikasi sms	

kepada semua kontak yang tersimpan jika pengguna terdeteksi
jatuh 60

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Orang yang sudah berusia lanjut telah banyak mengalami perubahan baik fisik maupun mental. Kedua perubahan ini sangat membawa dampak yang besar terhadap keamanan dan keselamatan bagi lansia. Salah satu masalah yang sering terjadi adalah ketidakstabilan. Masalah ketidakstabilan ini antara lain adalah lansia sering mengalami masalah jatuh. Jatuh pada lansia sendiri merupakan masalah yang sering terjadi dan memiliki banyak penyebab, baik faktor intrinsik misalnya gangguan sistem susunan syaraf, gangguan sistem anggota gerak, gangguan penglihatan dan pendengaran atau faktor ekstrinsik misalnya cahaya ruangan yang kurang terang, lantai licin, lingkungan yang asing bagi lansia [1]. Akibatnya, lansia yang jatuh akan sangat mudah mengalami cedera seperti patah tulang, luka pada kulit, kerusakan pada jaringan ikat dan jaringan lunak, bahkan yang terburuk bisa mengakibatkan kematian.

Berbagai penelitian telah dilakukan untuk mengamati dan menganalisa gerak jatuh seseorang agar dapat mengurangi dampak jatuh yang ditimbulkan. Penelitian yang sudah ada untuk menentukan apakah orang tersebut jatuh atau tidak adalah menggunakan sensor *accelerometer* yang akan menghitung perbandingan akselerasi dengan *Activity of Daily Living* (ADL) [2] [3]. Akan tetapi, jika hanya menggunakan sensor *accelerometer* saja untuk menentukan jatuh atau tidaknya seseorang, kemungkinan salah saat mendeteksi jatuh akan lebih besar seperti pada kondisi seseorang sedang duduk dengan cepat, melompat.

Penelitian lain adalah sistem pengenalan postur untuk menentukan seseorang jatuh atau dalam kondisi postur normal (tidak sedang jatuh) yang berbasis pada ekstraksi fitur dari sinyal kamera 2D [4]. Namun sistem tersebut tidak akan selalu bekerja secara maksimal karena ada faktor yang akan mengganggu kualitas sinyal gambar 2D seperti cahaya yang ditangkap oleh kamera. Selain itu kamera hanya dapat diletakkan di tempat tertentu dan tidak akan bisa memonitor seluruh aktivitas yang dilakukan seseorang.

Pada tugas akhir ini diusulkan sistem pendeteksi jatuh dan pemberitahuan dengan menggunakan *multisensor* berbasis *accelerometer* dan *gyroscope*. Sistem akan dibangun menggunakan dua sensor *accelerometer* dan *gyroscope* yang dipasang pada dada bagian kiri dan paha bagian kiri pengguna untuk mengambil data sampel dari gerakan pengguna. Algoritma yang diimplementasikan pada tugas akhir ini diadaptasi dari [5] [6] yang kemudian disesuaikan dengan kebutuhan. Selain itu *microcontroller arduino* akan dilengkapi dengan modul *bluetooth* untuk dapat berkomunikasi dengan perangkat android.

Metode yang diusulkan adalah dengan membedakan posisi pengguna ketika berdiri dan duduk. Kemungkinan jatuh pengguna pada saat duduk lebih kecil dibandingkan pada saat berdiri. Dengan membedakan posisi pengguna diharapkan dapat meminimalkan kesalahan hasil deteksi dari sistem *fall detection*, metode tersebut yang selanjutnya akan dinamakan dengan *quad threshold*.

Sistem ini diharapkan dapat menghasilkan akurasi yang tinggi untuk mendeteksi jatuh pada seseorang. Jika pengguna terdeteksi jatuh, sistem ini akan memberikan pemberitahuan kepada keluarga atau kerabat pengguna berupa data dan lokasi pengguna yang dikirimkan melalui sms.

1.2 Rumusan Masalah

Berikut beberapa hal yang menjadi rumusan masalah pada Tugas Akhir ini:

1. Bagaimana cara mendeteksi seseorang yang tiba – tiba jatuh pada saat melakukan aktivitas sehari – hari dengan menggunakan sensor *accelerometer* dan *gyroscope*?
2. Bagaimana mengatur *synchronization* antar dua modul sensor *accelerometer* dan *gyroscope*?
3. Bagaimana mekanisme pengiriman data dari *microcontroller* arduino ke *smartphone*?
4. Bagaimana tingkat akurasi dari sistem yang dibangun?

1.3 Batasan Masalah

Batasan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut:

1. Sistem ini menggunakan dua sensor *accelerometer* dan *gyroscope* yang diaplikasikan di bagian tubuh pengguna.
2. Sistem ini menggunakan bahasa pemrograman C untuk *microcontroller arduino* dan Java untuk pemrograman android.
3. Sistem ini digunakan pada saat pengguna melakukan aktivitas sehari – hari di rumah.
4. Sistem ini menggunakan perangkat android dengan versi minimal 5.0 dan sudah terintegrasi dengan *bluetooth low energy* dan GPS.
5. Notifikasi pada sistem ini menggunakan sms.

1.4 Tujuan Tugas Akhir

Tujuan dari tugas akhir ini adalah untuk membuat *fall detection* pada pengguna dan sistem notifikasi dengan analisis data sensor *accelerometer* dan *gyroscope*.

1.5 Manfaat Tugas Akhir

Tugas akhir ini diharapkan dapat memberikan informasi kepada keluarga dari pengguna apabila pengguna mengalami jatuh, terutama pada pengguna yang sudah lanjut usia. Penggunaan modul sensor *accelerometer* dan *gyroscope* yang berjumlah lebih dari satu diharapkan dapat memberikan hasil akurasi sistem yang lebih baik.

1.6 Metodologi

Pembuatan tugas akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.1.1 Penyusunan Proposal

Tahap awal penyusunan tugas akhir ini adalah penyusunan proposal tugas akhir. Pada proposal tugas akhir berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

1.1.2 Studi Literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai

Microcontroller Arduino, Arduino Software (IDE), Sensor Accelerometer, Sensor Gyroscope, Bluetooth, dan Android Studio.

1.1.3 Perancangan dan Desain Sistem

Analisis kebutuhan dan perancangan sistem dilakukan untuk merumuskan solusi yang tepat dalam pembuatan sistem serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan sesuai dengan studi literatur. Tahap ini mendefinisikan alur dari implementasi, langkah – langkah yang dikerjakan dan pembuatan rancangan dasar dari sistem.

1.1.4 Implementasi Sistem

Implementasi merupakan tahap untuk merealisasikan perancangan dan desain yang telah dibuat. Data dan fungsi yang ada pada tahap implementasi dibuat berdasarkan hasil dari perancangan dan desain yang telah dirumuskan pada tahap perancangan dan desain sistem.

1.1.5 Pengujian dan Evaluasi

Pengujian dilakukan pada sukarelawan yang melakukan aktivitas sehari – hari atau *Activity of Daily Living* (ADL) dan memperagakan gerak jatuh pada matras yang telah disediakan. Dilakukan berbagai gerakan seperti duduk, merebahkan badan, dan gerakan jatuh untuk mengetahui seberapa jauh akurasi sistem dalam membedakan ADL dengan jatuh.

1.1.6 Penyusunan Buku

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi yang telah dibuat.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian sebagai berikut:

Bab I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang, rumusan permasalahan, Batasan permasalahan, tujuan, manfaat, metodologi, dan sistematika penulisan laporan tugas akhir.

Bab II Tinjauan Pustaka

Bab ini berisi tentang penjelasan mengenai beberapa penunjang yang akan digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Perancangan dan Desain Sistem

Bab ini berisi pembahasan mengenai perancangan dan desain sistem yang selanjutnya akan digunakan untuk implementasi dalam pembuatan tugas akhir ini.

Bab IV Implementasi

Bab ini menjelaskan tentang penerapan dan realisasi dari bab sebelumnya dan pengambilan data – data sebagai dasar pembuatan tugas akhir ini.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan tahapan uji coba dan hasil uji coba yang telah dilakukan dalam pembuatan tugas akhir ini.

Bab VI Kesimpulan dan Saran

Bab ini berisikan kesimpulan yang diambil dari hasil pengujian dan evaluasi dan saran yang didapatkan dari permasalahan yang didapatkan pada pembuatan tugas akhir ini.

BAB II

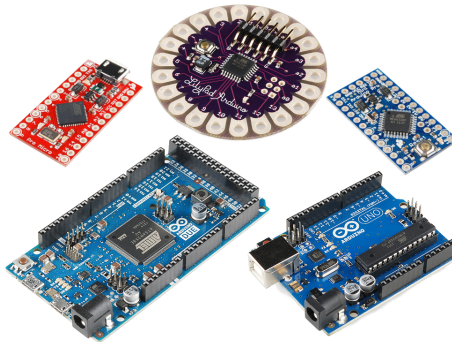
TINJAUAN PUSTAKA

Pada bab ini diuraikan mengenai detail dari penunjang dan dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan.

2.1 *Microcontroller Arduino*

Arduino adalah platform *open source* yang digunakan untuk membangun proyek elektronik. *Arduino* terdiri dari *physical programmable circuit board* yang sering disebut *microcontroller* beserta *software* atau IDE (*Integrated Development Environment*) yang digunakan untuk membuat kode programnya.

Arduino lahir di Ivrea Interaction Design Institute sebagai alat yang sangat mudah untuk mempercepat *prototyping*. *Arduino* tidak memerlukan *hardware* yang terpisah untuk mengaplikasikan *code* dari IDE ke *board*, hanya diperlukan kabel USB saja untuk menghubungkannya. Beberapa pertimbangan untuk menggunakan *arduino* seperti murah, *cross-platform*, simpel dan mudah digunakan dalam *programming*, *open source* baik *software* maupun *hardware*-nya [7] [8].



Gambar 2.1 Microcontroller Arduino [9]

2.2 Arduino Software (IDE)

Arduino Integrated Development Environment atau biasa disebut *Arduino Software (IDE)* berisi *text editor* untuk menulis *code*, area pesan, area *text console*, *toolbar* dengan tombol untuk beberapa fungsi umum dan serangkaian menu. *Arduino Software (IDE)* bisa disambungkan dengan *hardware arduino* dan *genuino* untuk mengupload program dan berkomunikasi dengan *board* tersebut [10].



Gambar 2.2 Arduino Software

2.3 Sensor Accelerometer

Accelerometer adalah sebuah sensor yang digunakan untuk mengukur kecepatan suatu objek. *Accelerometer* berfungsi mengukur percepatan *dynamic* dan *static*. Pengukuran *dynamic* adalah pengukuran percepatan pada objek bergerak, sedangkan

pengukuran *static* adalah pengukuran terhadap gravitasi bumi untuk mengukur sudut kemiringan suatu benda saat digerakkan.

Prinsip kerja yang digunakan adalah prinsip percepatan (*acceleration*). Beban bergerak dengan suatu percepatan sampai kondisi tertentu akan berhenti. Bila ada sesuatu yang menggoncangkannya maka beban akan berayun kembali. Pengukuran kapasitas inilah yang umumnya menjadi hasil pengukuran *chip*. Agar sensor bisa mendeteksi 3 dimensi, maka dibutuhkan 3 pasang plat yang dipasang tegak lurus antar masing – masing sudut sehingga sensor ini bisa mendeteksi gerak [11].

2.4 Sensor Gyroscope

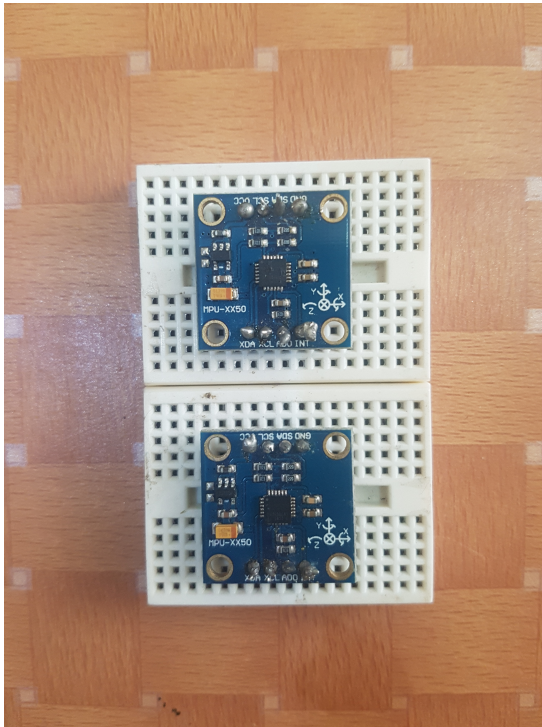
Gyroscope atau *Gyro* adalah perangkat untuk mengukur atau mempertahankan orientasi dengan prinsip ketetapan momentum sudut. Mekanismenya adalah sebuah roda berputar dengan piringan di dalamnya yang tetap stabil. *Gyroscope* dahulu sering digunakan pada robot atau heli dan alat – alat canggih lainnya namun sekarang perangkat *smartphone* pun tak mau kalah dan mulai menggunakannya. *Gyroscope* memiliki keluaran berupa kecepatan sudut dari arah 3 sumbu yaitu: sumbu x yang nantinya akan menjadi sudut *phi* (kanan dan kiri) dari sumbu y nantinya menjadi sudut *tetha* (atas dan bawah), dan sumbu z nantinya menjadi sudut *psi* (depan dan belakang) [12].

2.5 Modul GY – 52 MPU – 6050

Perangkat MPU – 6050 adalah perangkat *motion tracking* pertama yang didesain untuk kebutuhan berdaya rendah, biaya rendah, dan dengan kinerja tinggi. Perangkat MPU – 6050 berisi *accelerometer* MEMS (*Micro Electro Mechanical System*) dan *gyroscope* MEMS dalam satu *chip* yang berisi *hardware converter* 16 – bit analog ke digital pada setiap channelnya. Perangkat ini menggunakan I2C bus untuk berinteraksi dengan *microcontroller* *arduino* [13]. Setiap sensor *accelerometer* dan *gyroscope* terdiri

dari tiga sumbu putar x, y, dan z. Pada sensor *accelerometer* memiliki rentang skala yang dapat disesuaikan hingga +2g, +4g, +8g, dan +16g. Sedangkan pada sensor *gyroscope* memiliki rentang skala yang dapat disesuaikan dengan rentang +250, +500, +1000, dan +2000 derajat per detik (dps) [14].

Pada MPU – 6050 ada beberapa “*breakout boards*” atau “*sensor boards*”, salah satunya adalah GY – 52. *Sensor board* ini memiliki pengatur tegangan yang berfungsi untuk mengatur tegangan apakah berada pada tegangan 5V atau 3,3V. *Sensor board* ini awalnya dirancang untuk perangkat MPU – 3050, sehingga kadang tertulis tulisan “MPU – 3050” atau “MPU – X050” pada *sensor board* [13].



Gambar 2.3 Modul GY - 52

2.6 Bluetooth

Bluetooth adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping transceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real – time* antara *host – host bluetooth* dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter). *Bluetooth* sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan *card* yang digunakan untuk *wireless local area network (WLAN)* dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada *bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah.

Pada dasarnya *bluetooth* diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan kabel di dalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi *mobile wireless* dengan biaya yang relatif rendah, konsumsi daya yang rendah, *interoperability* yang menjanjikan, mudah dalam pengoperasian dan mampu menyediakan layanan yang bermacam – macam [15].

2.7 Android

Android adalah sebuah sistem operasi berbasis *linux* yang dirancang khusus untuk perangkat selular / telepon genggam yang biasa disebut sebagai gadget atau *smartphone* (telepon pintar) dan *computer tablet* (tab). Android awalnya dikembangkan oleh Android inc, sebuah perusahaan yang membuat *software* untuk ponsel yang berada di Palo Alto, California Amerika Serikat. Di tahun 2005 Google membeli/mengakuisisi perusahaan yang waktu itu bernama Android, sebuah perusahaan yang berkecimpung di dunia *mobile* [16].

Keunggulan utama Android adalah gratis dan *open source*, yang membuat *smartphone* Android dijual lebih murah

dibandingkan dengan Blackberry atau iPhone meski fitur *hardware* yang ditawarkan android lebih baik [17]. Perjalanan android pun sudah sampai ke generasi ke delapan dalam sepuluh tahun terakhir. Berikut versi android dari pertama kali diluncurkan sampai versi yang terbaru:

1. Android 1.0
2. Android 1.1
3. Android 1.5 (*Cupcake*)
4. Android 1.6 (*Donut*)
5. Android 2.1 (*Eclair*)
6. Android 2.2 (*Froyo*)
7. Android 2.3 (*Gingerbread*)
8. Android 3.0 (*Honeycomb*)
9. Android 4.0 (*Ice Cream Sandwich*)
10. Android 4.1 (*Jelly Bean*)
11. Android 4.4 (*Kitkat*)
12. Android 5.0 (*Lollipop*)
13. Android 6.0 (*Marshmallow*)
14. Android 7.0 (*Nougat*)
15. Android 8.0 (*Oreo*)

2.8 Android Studio

Android Studio adalah *official Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android. Android Studio merupakan suatu pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi android. Android Studio menggunakan *gradle* sebagai *build environment*. Fitur yang sudah disematkan pada android Studio [18] [19], antara lain :

- *Gradle – based build system* yang fleksibel
- Emulator yang cepat dan memiliki banyak fitur
- Dapat digunakan untuk mengembangkan semua *device* android

- Fitur *Instant Run* untuk mengupdate perubahan tanpa harus *build file apk* baru
- Dukungan C++ dan NDK
- Dan masih banyak lagi



Gambar 2.4 Android Studio

[Halaman ini sengaja dikosongkan]

BAB III

PERANCANGAN DAN DESAIN SISTEM

Pada bab ini diuraikan mengenai perancangan aplikasi agar dapat mencapai tujuan dari tugas akhir. Perancangan merupakan bagian penting yang berupa perencanaan – perencanaan secara teknis sistem yang akan dibuat. Dimulai dari deskripsi umum sistem hingga perancangan proses, alur dan implementasinya.

3.1 Deskripsi Umum Sistem

Pada tugas akhir ini akan dibangun suatu sistem *fall detection* yang ditujukan untuk pengguna dengan menggunakan *microcontroller arduino* yang dilengkapi dengan 2 sensor *accelerometer* dan 2 sensor *gyroscope*. Sistem *fall detection* ini dirancang untuk melakukan *monitoring* terhadap aktivitas sehari – hari atau *Activity of Daily Living* (ADL) pada pengguna dan akan memberikan notifikasi jika pengguna terindikasi jatuh. Data yang dibutuhkan untuk menentukan jatuh atau tidaknya seseorang, yaitu data *accelerometer* dan *gyroscope*.

Data yang sudah didapatkan dari sensor *accelerometer* dan *gyroscope* kemudian diolah untuk mendapatkan nilai yang akan digunakan dalam menentukan kondisi jatuh pengguna. Data dari sensor *accelerometer* didapatkan nilai percepatan linear dan dari sensor *gyroscope* didapatkan kecepatan sudut yang masing – masing memiliki sumbu x, y, z . Selanjutnya data dari sensor *accelerometer* disimbolkan menjadi ax, ay, az yang merepresentasikan nilai percepatan linear pada setiap sumbu dan data dari sensor *gyroscope* disimbolkan menjadi $\omega x, \omega y, \omega z$ yang merepresentasikan nilai kecepatan sudut pada setiap sumbu.

Nilai yang telah didapatkan tersebut kemudian dihitung untuk mendapatkan nilai resultan percepatan sudut a_m , selisih percepatan linear Δa_m , resultan kecepatan sudut ω_m , selisih resultan kecepatan sudut $\Delta \omega_m$, dan perubahan sudut θ_m . Berikut

adalah perhitungan untuk mendapatkan nilai – nilai yang sudah disebutkan sebelumnya:

$$a_m = \sqrt{ax_m^2 + ay_m^2 + az_m^2} \quad (3.1)$$

$$\omega_m = \sqrt{\omega x_m^2 + \omega y_m^2 + \omega z_m^2} \quad (3.2)$$

$$\Delta a_m = \max(a_{m[1]} \dots a_{m[n-1]}) - \min(a_{m[1]} \dots a_{m[n-1]}) \quad (3.3)$$

$$\Delta \omega_m = \max(\omega_{m[1]} \dots \omega_{m[n-1]}) - \min(\omega_{m[1]} \dots \omega_{m[n-1]}) \quad (3.4)$$

$$\theta_m = \arccos \frac{a_m}{g} \quad (3.5)$$

Nilai – nilai yang sudah didapatkan dari persamaan tersebut diatas selanjutnya akan dibandingkan dengan nilai *threshold* yang sudah ditentukan untuk dapat membedakan antara jatuh dan kegiatan sehari – hari. Nilai *threshold* didapatkan dari pengambilan data *training* berupa percobaan memperagakan beberapa gerakan jatuh dan gerakan saat melakukan aktivitas sehari – hari.

Secara umum perancangan sistem akan dibagi menjadi 3 proses utama, yaitu:

1. Perancangan *microcontroller arduino* dengan integrasi sensor *accelerometer* dan *gyroscope* untuk mengambil data yang dibutuhkan.
2. Perancangan pengolahan data yang didapatkan oleh sensor *accelerometer* dan *gyroscope* pada *microcontroller arduino* untuk menghasilkan nilai percepatan linear, kecepatan sudut, resultan percepatan linear, resultan kecepatan sudut, dan perubahan sudut untuk menentukan apakah pengguna jatuh atau tidak.
3. Perancangan aplikasi untuk memberikan notifikasi pada keluarga atau kerabat pengguna menggunakan perangkat android.

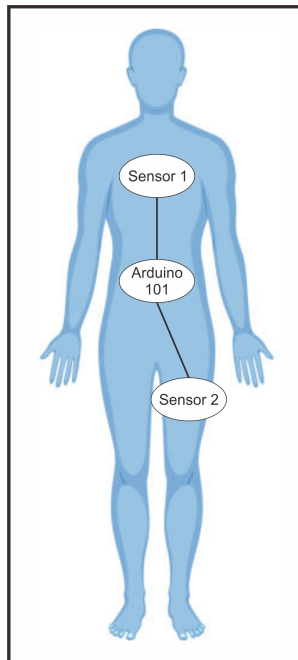
3.2 Arsitektur Umum Sistem

Sistem ini melibatkan beberapa komponen, dan setiap komponen memiliki peran masing – masing. Komponen – komponen tersebut adalah sebagai berikut:

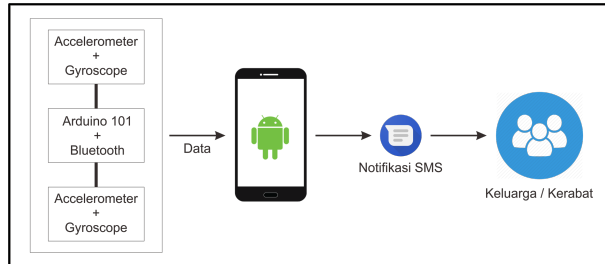
1. Pengguna
Pengguna adalah seseorang yang menggunakan pakaian yang dilengkapi dengan *microcontroller arduino*, dan juga merupakan pengguna perangkat android.
2. Perangkat Sensor
Perangkat sensor berperan pada proses pengambilan data yang selanjutnya akan diolah pada *microcontroller arduino*. Perangkat sensor ini terdiri dari sensor *accelerometer* dan sensor *gyroscope* yang sudah terintegrasi pada modul GY – 52.
3. *Microcontroller Arduino*
Microcontroller arduino berperan untuk mengambil data dari sensor *accelerometer* dan *gyroscope* untuk diolah sehingga dapat menentukan apakah pengguna terdeteksi jatuh atau tidak kemudian dikirimkan pada perangkat android melalui *bluetooth*.
4. Perangkat Android
Perangkat android berperan untuk menerima data dari *microcontroller arduino* yang kemudian digunakan untuk sarana mengirimkan notifikasi kepada keluarga atau kerabat pengguna.

Dari komponen – komponen diatas, dirancang sebuah arsitektur umum dari sistem yang dibangun dan diilustrasikan pada **Gambar 3.2.** *Microcontroller arduino* yang sudah terintegrasi dengan *bluetooth module* dirangkai dengan modul GY – 52 yang kemudian dihubungkan dengan *power bank* sebagai daya. Terdapat 2 modul GY – 52 yang dirangkai pada *microcontroller arduino*. *Microcontroller arduino* yang sudah tersambung dengan modul

GY – 52 dipasang pada pakaian pengguna, sensor pertama diletakkan di bahu sebelah kiri pengguna, sedangkan sensor kedua diletakkan di paha sebelah kiri pengguna. Setelah selesai pemasangan, sensor mulai mengambil data *accelerometer* dan *gyroscope*. Pengolahan data dilakukan pada *microcontroller arduino* untuk mendapatkan nilai percepatan linear, kecepatan sudut, resultan percepatan linear, resultan kecepatan sudut, dan perubahan sudut. Penentuan apakah pengguna jatuh atau tidak juga dilakukan pada *microcontroller arduino*. Jika didapatkan hasil jatuh maka *microcontroller arduino* akan mengirimkan status jatuh ke perangkat android melalui *bluetooth*. Perangkat android kemudian mengirimkan notifikasi jatuh pada keluarga/kerabat pengguna.



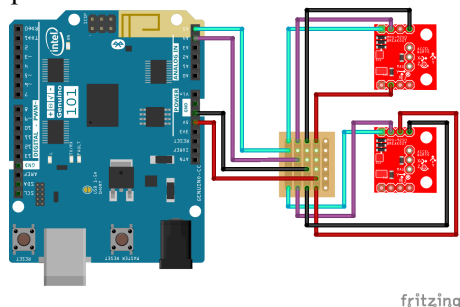
Gambar 3.1 Posisi penempatan sensor dan *microcontroller arduino*



Gambar 3.2 Arsitektur Umum Sistem

3.3 Perancangan Integrasi *Microcontroller Arduino* dengan Perangkat Sensor

Sensor *accelerometer* dan sensor *gyroscope* merupakan salah satu komponen utama pada sistem yang sudah terdapat pada modul GY – 52. Sensor tersebut akan digunakan untuk mengambil data yang dibutuhkan dalam proses *monitoring* kegiatan yang dilakukan oleh pengguna. Data yang diperoleh dari sensor tersebut kemudian diolah pada *microcontroller arduino* yang berperan sebagai otak dalam sistem. Untuk mengirim hasil jika terdeteksi pengguna jatuh, digunakan modul *bluetooth* yang sudah terintegrasi pada *microcontroller arduino*. Perancangan alat diilustrasikan pada **Gambar 3.3**.



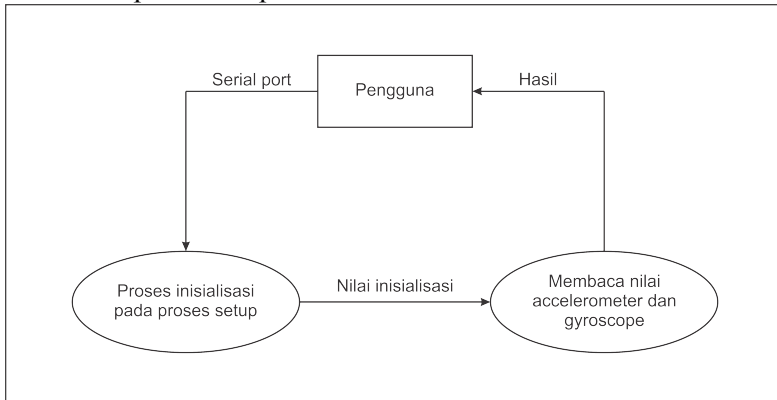
Gambar 3.3 Rangkaian integrasi *microcontroller arduino* dengan perangkat sensor

3.4 Perancangan Diagram Alir

Pada bagian ini akan dijelaskan alur kerja setiap proses yang ada pada sistem. Alur kerja pada proses yang ada pada sistem ini digambarkan pada diagram alir untuk mempermudah pemahaman secara garis besar proses yang ada pada sistem. Diagram alir pada sistem ini terdiri dari beberapa proses.

3.4.1 Inisialisasi *Microcontroller Arduino*

Ketika *arduino* dinyalakan, *arduino* akan menjalankan 2 fungsi utama dari *script* yang sudah diupload, yaitu fungsi *setup* dan *loop*. Fungsi *setup* hanya dijalankan sekali pada saat *aduno* pertama kali dijalankan. Fungsi *setup* ini berisi inisialisasi untuk konfigurasi *arduino* itu sendiri dan konfigurasi modul – modul yang akan dipasang pada *arduino* tersebut. Diagram alir inisiasi *arduino* dapat dilihat pada **Gambar 3.4**.

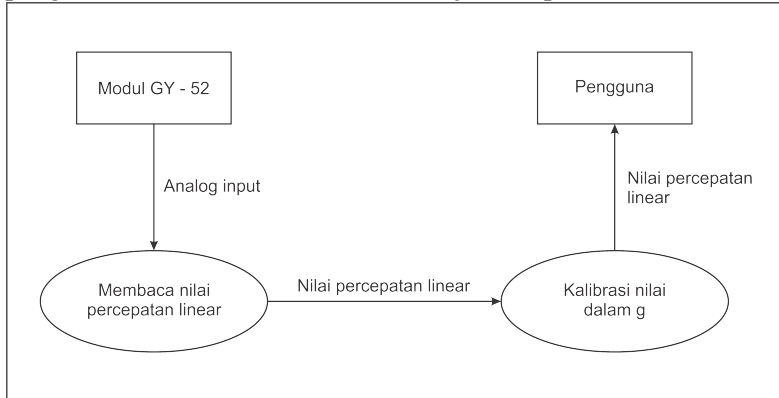


Gambar 3.4 Diagram alir inisialisasi *microcontroller arduino*

3.4.2 Pembacaan Nilai *Accelerometer*

Nilai percepatan linear yang diperoleh dari sensor *accelerometer* pada modul GY – 52 berupa nilai digital dalam

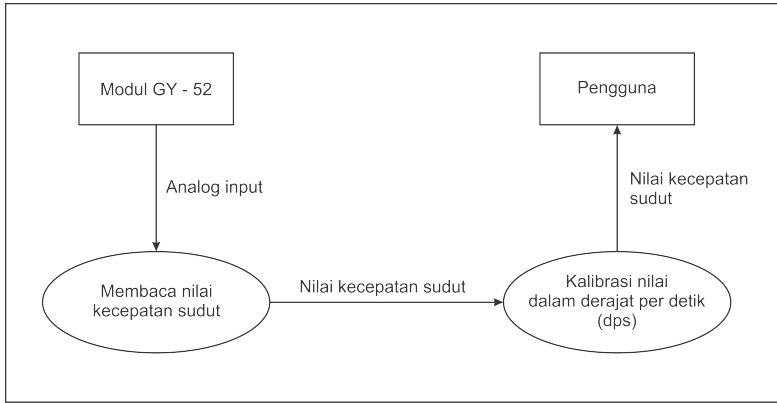
satuan bit. Nilai tersebut harus dikalibrasi untuk mendapatkan nilai yang memiliki *range* ± 2 dan dalam satuan gaya gravitasi (*g*). Pembacaan data dapat dilakukan secara berulang dengan menggunakan *delay* waktu tertentu. Diagram alir untuk pengambilan data *accelerometer* ditunjukkan pada **Gambar 3.5**.



Gambar 3.5 Diagram alir pembacaan nilai *accelerometer*

3.4.3 Pembacaan Nilai *Gyroscope*

Nilai kecepatan sudut yang diperoleh dari sensor *gyroscope* pada modul GY – 52 berupa nilai digital dalam satuan bit. Nilai tersebut harus dikalibrasi untuk mendapatkan nilai yang memiliki *range* ± 250 dan dalam satuan derajat/detik ($^{\circ}/s$). Pembacaan data dapat dilakukan secara berulang dengan menggunakan *delay* waktu tertentu. Diagram alir untuk pengambilan data *gyroscope* ditunjukkan pada **Gambar 3.6**.



Gambar 3.6 Diagram alir pembacaan nilai *gyroscope*

3.4.4 Algoritma *Fall Detection* Menggunakan Metode *Quad Threshold*

Sebelumnya sudah dijelaskan mengenai nilai – nilai yang telah didapatkan melalui pengambilan data dari sensor *accelerometer* maupun sensor *gyroscope*. Algoritma ini dirancang untuk mendeteksi aktivitas jatuh pada pengguna pada saat melakukan kegiatan sehari – hari. Algoritma ini memanfaatkan data dari sensor *accelerometer* yang berupa nilai percepatan linear pada setiap sumbu ax, ay, az dan sensor *gyroscope* yang berupa nilai kecepatan sudut pada setiap sumbu $\omega x, \omega y, \omega z$ yang kemudian diolah dan didapatkan nilai resultan percepatan linear a_m pada persamaan (3.1), resultan percepatan gravitasi ω_m pada persamaan (3.2), dan perubahan sudut θ_m pada persamaan (3.5). Pada awalnya, dideteksi terlebih dahulu apakah pengguna berada pada posisi berdiri atau tidak sedang berdiri dengan membandingkan nilai $\Delta\omega_2$ yang didapatkan dengan persamaan (3.4) dengan nilai *threshold* tw_2 dan nilai ay_2, az_2 dengan nilai *threshold* tay_2, taz_2 selama selang waktu 2 detik. Selanjutnya apabila didapatkan hasil bahwa pengguna sedang berdiri, dari nilai

selisih resultan percepatan linear Δa_1 yang didapatkan dengan persamaan (3.3), resultan kecepatan sudut $\Delta \omega_1$ yang didapatkan dengan persamaan (3.4), perubahan sudut θ_1 dari persamaan (3.5) didapatkan nilai threshold yang akan dibandingkan untuk membedakan antara jatuh dengan kegiatan sehari – hari. Diagram alir algoritma *fall detection* ditunjukkan pada **Gambar 3.7**.

3.4.5 Klasifikasi Posisi Berdiri

Pada bagian ini akan dijelaskan proses klasifikasi untuk menentukan apakah pengguna dalam keadaan berdiri atau tidak sedang berdiri. Dimulai dengan pembacaan nilai *analog input* dari sensor *accelerometer* dan *gyroscope* pada *microcontroller arduino*. Data tersebut kemudian dikalibrasi sehingga didapat nilai percepatan linear yang memiliki satuan g dan nilai kecepatan sudut yang memiliki satuan $^\circ/s$. Selanjutnya dari nilai percepatan linear diambil ay_2 , az_2 dan nilai kecepatan sudut diolah sehingga didapat nilai selisih resultan kecepatan sudut $\Delta \omega_2$. Kemudian dari nilai – nilai tersebut diproses kembali dengan cara membandingkan nilai – nilai tersebut dengan threshold yang sudah ditentukan untuk mengklasifikasikan apakah pengguna sedang berdiri atau tidak. Jika terdeteksi bahwa pengguna sedang berdiri, maka sistem akan melakukan proses selanjutnya yaitu menentukan apakah pengguna sedang jatuh atau tidak. Diagram alir data klasifikasi posisi berdiri ditunjukkan pada **Gambar 3.8**.

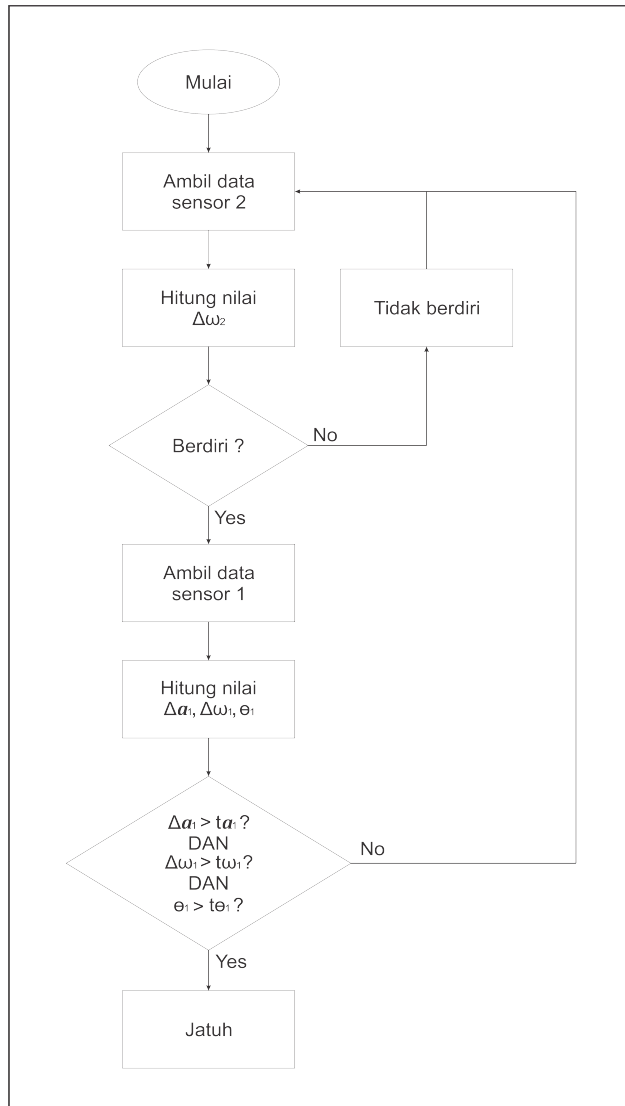
3.4.6 Penentuan Jatuh

Pada sub bab sebelumnya telah dijelaskan mengenai proses klasifikasi posisi berdiri. Jika sudah terdeteksi bahwa pengguna sedang berdiri, maka akan dilanjutkan dengan proses penentuan jatuh. Pada proses ini dilakukan kembali pembacaan nilai *analog input* dari sensor *accelerometer* dan *gyroscope* pada *microcontroller arduino*. Data yang didapatkan dikalibrasi sehingga didapatkan nilai percepatan linear dengan satuan g dan

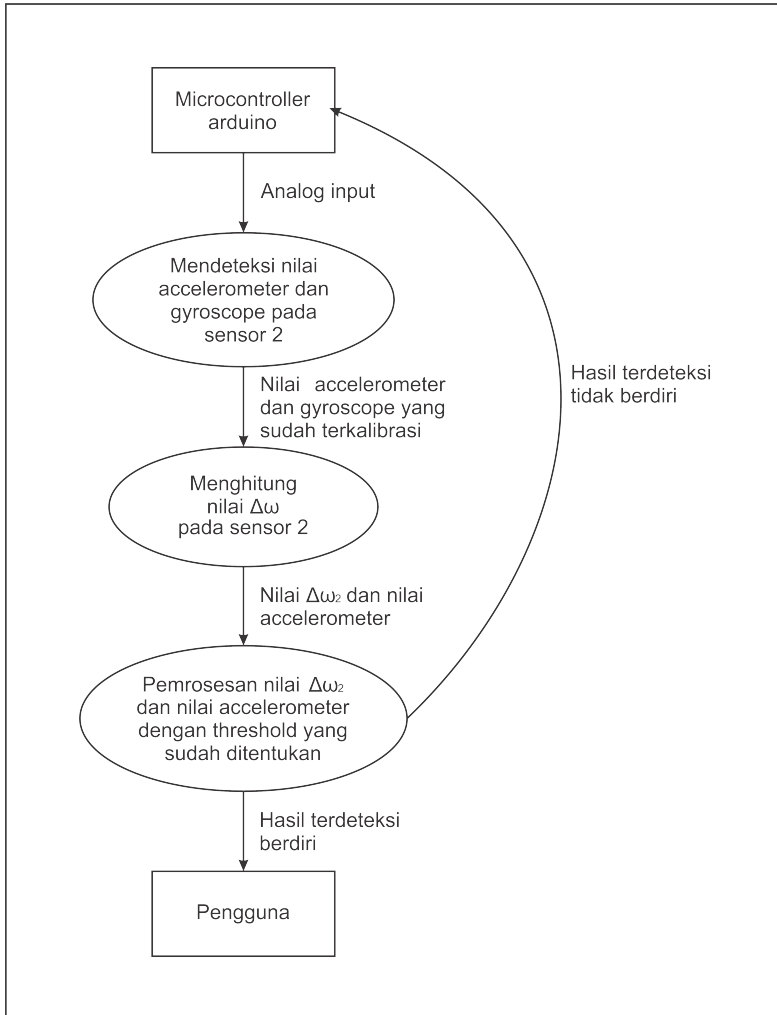
nilai kecepatan sudut dengan satuan $^{\circ}/s$. Selanjutnya dari nilai tersebut diolah sehingga didapatkan nilai selisih resultan percepatan linear Δa_1 , nilai selisih resultan kecepatan sudut $\Delta \omega_1$, nilai perubahan sudut pada sumbu x θ_x . Dari nilai tersebut didapatkan *threshold* yang kemudian dibandingkan untuk membedakan apakah pengguna sedang jatuh atau melakukan kegiatan sehari – hari. Diagram alir data penentuan jatuh ditunjukkan pada **Gambar 3.9**.

3.4.7 Pengiriman Notifikasi

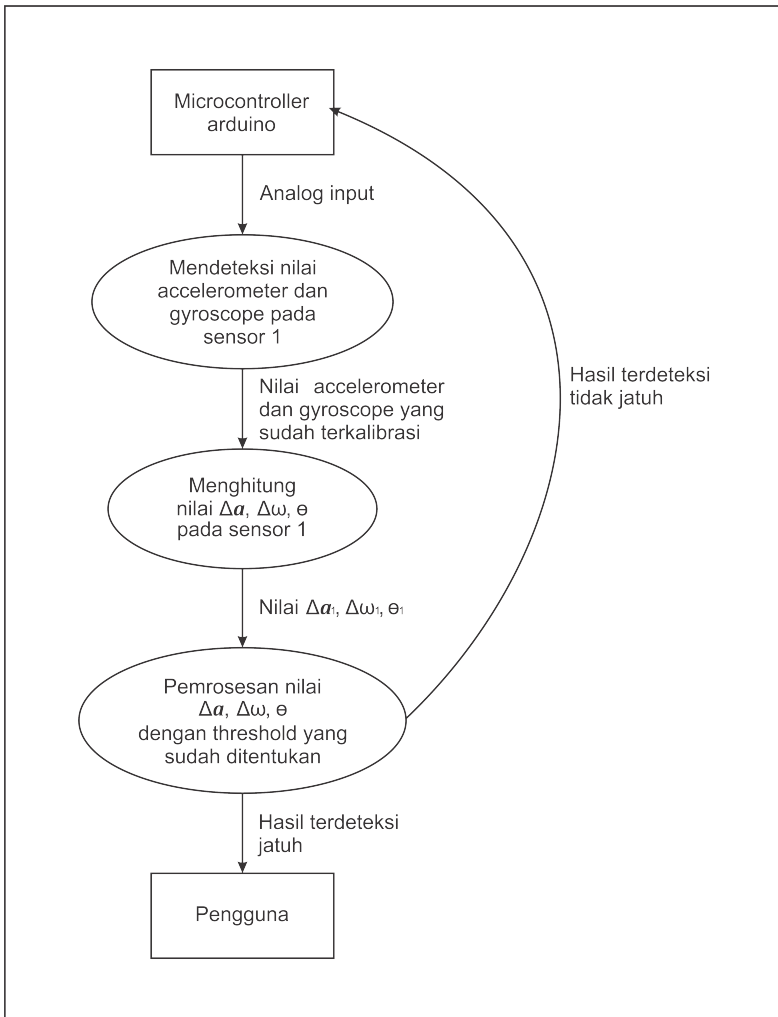
Pada bagian ini akan dijelaskan mengenai alur pengiriman notifikasi yang berupa sms kepada keluarga atau kerabat pengguna pada saat pengguna terdeteksi jatuh. Ketika terdeteksi bahwa pengguna sedang jatuh, *microcontroller arduino* akan mengirimkan data jatuh melalui *bluetooth* ke perangkat android milik pengguna yang sudah tersambung. Selanjutnya perangkat android tersebut menentukan letak posisi pengguna dengan menggunakan GPS. Setelah didapatkan posisinya, perangkat android akan mengirimkan notifikasi yang berisi pemberitahuan bahwa pengguna sedang jatuh dilengkapi dengan tautan lokasi jatuh pengguna melalui sms kepada keluarga atau kerabat pengguna. Diagram alir pengiriman notifikasi ditunjukkan pada **Gambar 3.10**.



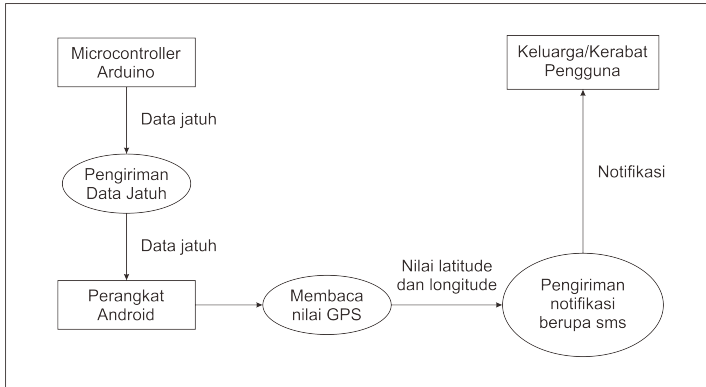
Gambar 3.7 Diagram alir algoritma *fall detetction* menggunakan metode *quad threshold*



Gambar 3.8 Diagram alir klasifikasi posisi berdiri



Gambar 3.9 Diagram alir penentuan jatuh



Gambar 3.10 Diagram alir pengiriman notifikasi

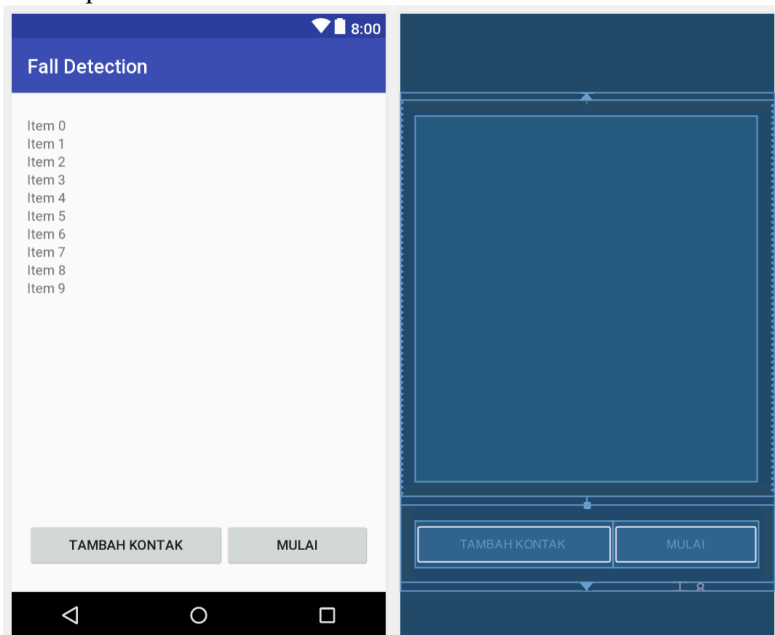
3.5 Perancangan Antarmuka Sistem

Perangkat android pada sistem ini berfungsi sebagai jembatan penghubung dari perangkat sensor untuk kemudian dilakukan pengiriman notifikasi kepada keluarga / kerabat pengguna. Data yang didapatkan oleh perangkat sensor yang selanjutnya diolah dengan *microcontroller arduino* yang menghasilkan data yang berupa notifikasi jatuh atau tidak kemudian dikirim ke perangkat android melalui komunikasi *bluetooth*. Setelah data yang menunjukkan gerak jatuh pengguna diterima oleh perangkat android, perangkat android akan menentukan lokasi pengguna dan kemudian mengirimkan sms notifikasi kepada keluarga / kerabat pengguna yang kontakannya terdaftar pada perangkat android.

Aplikasi pada perangkat android memiliki beberapa fitur utama, yang pertama menambah, mengubah, menghapus data kontak dari kerabat pengguna. Kedua, mencari perangkat *bluetooth* yang sudah terintegrasi dalam perangkat sensor. Ketiga, melakukan proses penerimaan data dan mengirimkan notifikasi berupa sms kepada keluarga / kerabat pengguna.

3.5.1 Rancangan Antarmuka Awal Aplikasi

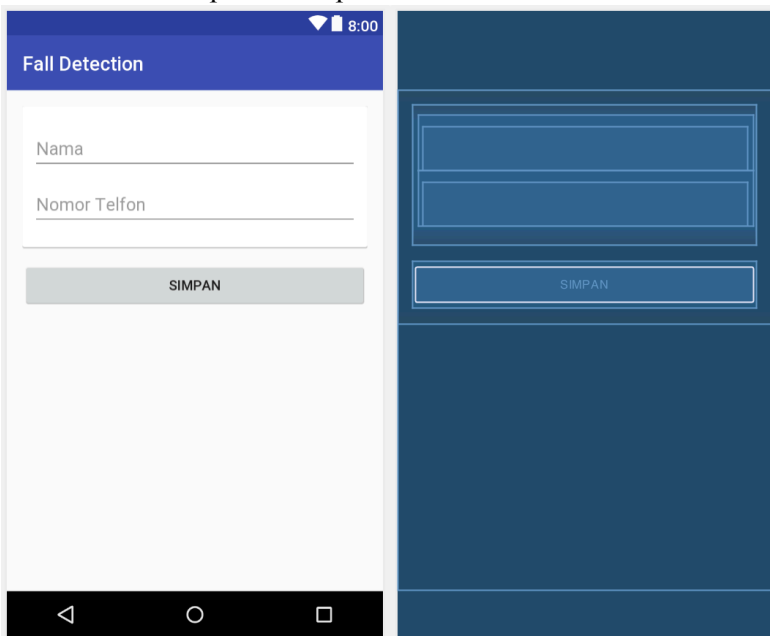
Rancangan antarmuka awal aplikasi merupakan rancangan antarmuka pada saat perangkat pertama kali menjalankan aplikasi. Menu utama pada aplikasi akan menampilkan daftar kontak yang sudah didaftarkan pada aplikasi ini. Selain daftar kontak, juga terdapat dua tombol, yaitu tombol tambah kontak untuk mendaftarkan kontak baru dan tombol mulai untuk memulai pencarian perangkat *microcontroller arduino* yang sudah terintegrasi *bluetooth*. Rancangan antarmuka awal aplikasi dapat dilihat pada **Gambar 3.11**.



Gambar 3.11 Rancangan antarmuka awal aplikasi

3.5.2 Rancangan Antarmuka Tambah Kontak

Jika pengguna menekan tombol tambah kontak pada halaman awal aplikasi, pengguna akan diarahkan ke halaman tambah kontak. Rancangan antarmuka ini berisi kolom *form* pengisian nama dan nomor telepon untuk menambahkan kontak. Selain itu terdapat tombol simpan yang berfungsi untuk menyimpan data kontak yang telah diisi. Rancangan antarmuka tambah kontak dapat dilihat pada **Gambar 3.12**.

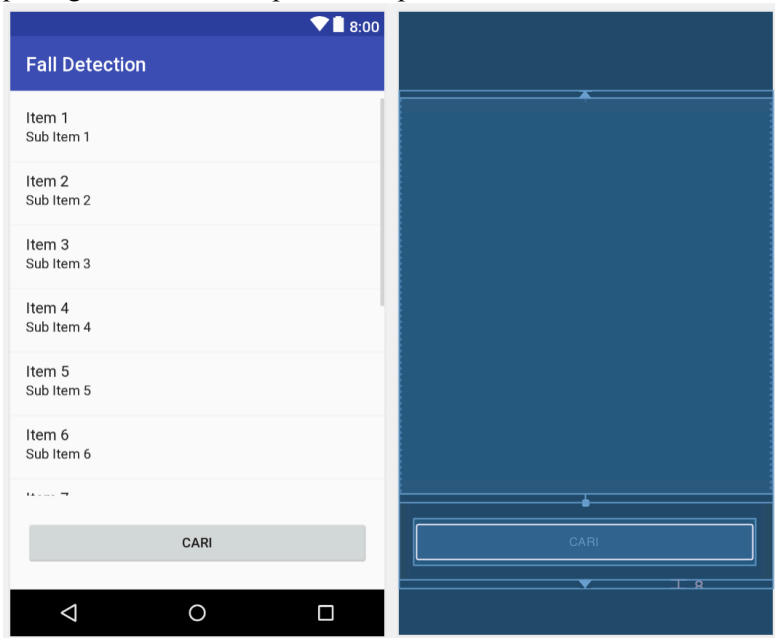


Gambar 3.12 Rancangan antarmuka tambah kontak

3.5.3 Rancangan Antarmuka Daftar Perangkat *Bluetooth*

Jika pengguna menekan tombol mulai pada halaman awal aplikasi, pengguna akan diarahkan ke halaman daftar perangkat *bluetooth*. Halaman ini berfungsi untuk mencari perangkat

microcontroller arduino yang telah terintegrasi *bluetooth*. Rancangan antarmuka ini berisi daftar perangkat yang memiliki *bluetooth* yang terdeteksi oleh aplikasi. Terdapat juga tombol cari yang berfungsi untuk mencari perangkat yang memiliki *bluetooth*. Defaultnya aplikasi akan secara otomatis mencari perangkat yang memiliki *bluetooth*, tetapi jika perangkat *bluetooth* tidak terdeteksi maka pengguna dapat kembali menekan tombol cari untuk mencari perangkat *bluetooth* yang diinginkan. Rancangan antarmuka daftar perangkat *bluetooth* dapat dilihat pada **Gambar 3.13**.

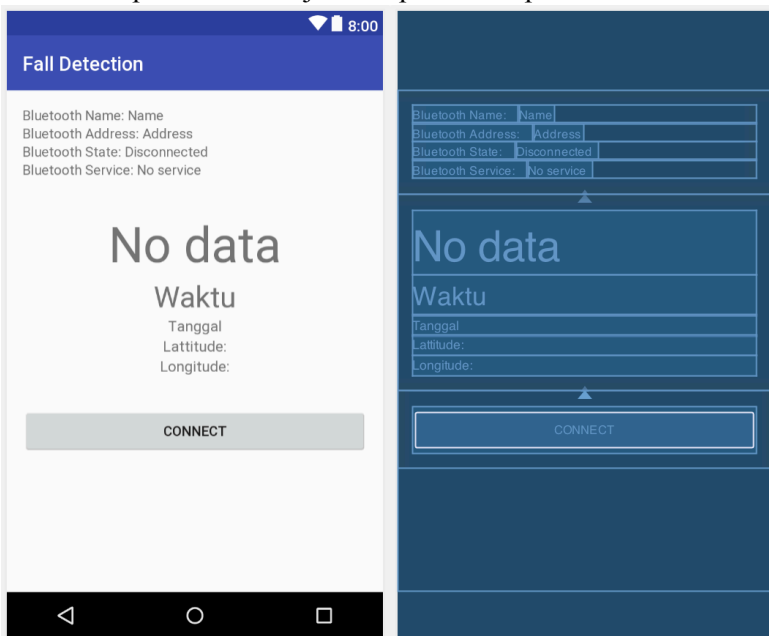


Gambar 3.13 Rancangan antarmuka daftar perangkat *bluetooth*

3.5.4 Rancangan Antarmuka Proses Deteksi Jatuh

Jika pengguna menekan salah satu item dalam daftar perangkat yang ditampilkan pada halaman daftar perangkat

bluetooth, maka pengguna akan diarahkan ke halaman proses deteksi jatuh. Rancangan antarmuka ini berisi data *bluetooth* seperti nama *bluetooth*, alamat *bluetooth*, koneksi *bluetooth*, dan *service* dari *bluetooth* tersebut. Selain itu ditampilkan juga data yang dikirimkan *microcontroller arduino* melalui *bluetooth* tersebut, jam, hari, tanggal dan keterangan *latitude* dan *longitude* dimana perangkat android berada. Juga ada tombol *connect* yang berfungsi untuk melakukan koneksi perangkat android ke perangkat *microcontroller arduino* melalui *bluetooth*. Rancangan antarmuka proses deteksi jatuh dapat dilihat pada **Gambar 3.14**.



Gambar 3.14 Rancangan antarmuka proses deteksi jatuh

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi sistem dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam aplikasi. Selain itu, implementasi dari perangkat keras dan beberapa keterangan yang berhubungan dengan sistem juga dijelaskan.

4.1 Lingkungan Implementasi

Pada penelitian ini, digunakan beberapa perangkat sebagai pendukung. Perangkat – perangkat tersebut dibagi menjadi dua bagian, yaitu lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak.

Tabel 4.1 Lingkungan implementasi

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Laptop MacBook Pro 13 – inch	Intel® Core i5 2.7 GHz, 8 Gb 1867 MHz DDR3 RAM
	Arduino 101	Intel® Curie 32 MHz (7 – 12V), 196 Kb Flash Memory, 24 Kb SRAM
	Smartphone Samsung Galaxy S7 Flat	Samsung Exynos Octa 8890 2.60 GHz, 4 Gb RAM

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Lunak	Sistem Operasi Laptop	macOS High Sierra 10.13.4
	Sistem Operasi Perangkat Android	Android 7.0 Nougat
	Arduino IDE	Arduino Development Kit versi 1.8.5
	Android SDK	Android Studio versi 3.1.2

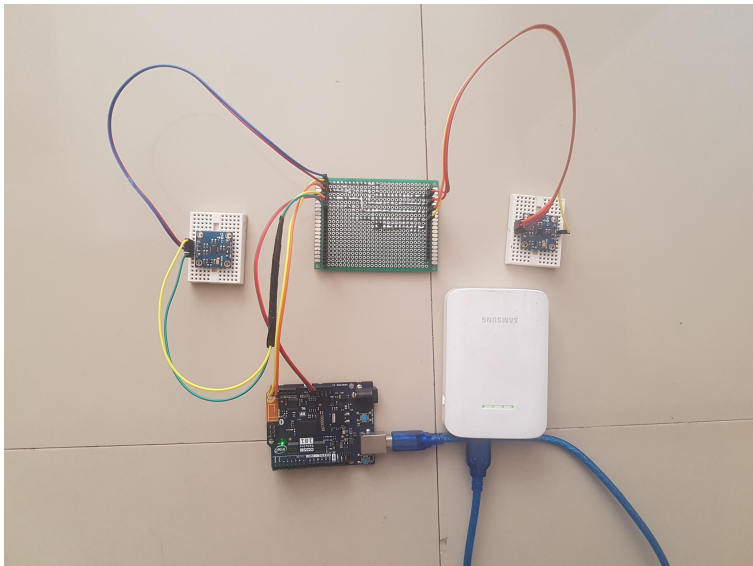
4.2 Implementasi Perangkat Keras

Pada sub bab implementasi ini menjelaskan mengenai pembangunan perangkat keras secara detail dan menampilkan *Source code* yang digunakan. Proses implementasi perangkat keras ini diawali dengan sebuah *prototype* untuk menguji apakah perangkat keras dapat berfungsi untuk pengambilan data yang dibutuhkan dalam sistem ini. Dari perancangan perangkat keras ini diharapkan *output* berupa hasil dari pengolahan data sensor *accelerometer* dan *gyroscope* dari modul GY – 52. Pada sistem ini, perangkat keras yang digunakan adalah sebagai berikut:

1. Satu buah *microcontroller arduino* 101
2. Dua buah modul GY – 52
3. Satu set kabel *jumper*
4. Satu buah PCB
5. Dua buah *breadboard* mini
6. Satu buah power bank

Microcontroller arduino berfungsi sebagai otak perangkat yang akan mengatur semua proses baik pengambilan data melalui sensor, pengolahan data, maupun pengiriman data melalui *bluetooth*. Dua buah modul GY – 52 terlebih dahulu dipasang pada masing – masing *breadboard* mini sebagaiudukan untuk kemudian dihubungkan ke *microcontroller arduino* dengan kabel *jumper*. Pada modul GY – 52 yang pertama digunakan empat *pin* yang akan dihubungkan ke *microcontroller arduino*, yaitu *pin GND*, *SDA*, *SCL*, dan *VCC*. Sedangkan pada modul GY – 52 yang kedua, yang digunakan untuk menghubungkan modul tersebut dengan *microcontroller arduino* adalah *pin GND*, *SDA*, *SCL*, dan *AD0*.

Rangkaian perangkat sensor tersebut kemudian dihubungkan dengan daya yang didapatkan dari power bank agar proses pengambilan, pemrosesan dan pengiriman data berjalan dengan lancar. Rangkaian perangkat sensor dapat dilihat pada **Gambar 4.1**.



Gambar 4.1 Rangkaian perangkat sensor

4.3 Implementasi pada *Microcontroller Arduino*

Pada bagian ini dijelaskan mengenai implementasi *script* yang akan diupload pada *microcontroller arduino* dimana akan digunakan untuk mengatur semua proses yang berkaitan dengan *microcontroller arduino* dan perangkat sensor pada sistem ini.

Penjelasan dari masing-masing tahapan dari proses tersebut adalah sebagai berikut:

4.3.1 Implementasi Inisialisasi *Microcontroller Arduino*

Proses inisialisasi pada *microcontroller arduino* dilakukan pertama kali dan hanya sekali pada saat *microcontroller arduino* dihidupkan. Proses tersebut dilakukan pada fungsi setup. Inisialisasi *microcontroller arduino* ditunjukkan pada gambar --- dengan tahapan sebagai berikut:

1. Inisialisasi I2C bus untuk membangun komunikasi antara *microcontroller arduino* dan modul GY – 52.
2. Membuka *port serial*.
3. Mengatur modul GY – 52 pada alamat 0x68 dan 0x69.
4. Inisialisasi modul GY – 52.
5. Melakukan tes koneksi pada modul GY – 52.
6. Mengatur nama *bluetooth* yang akan dideteksi oleh perangkat android.
7. Mengatur *service bluetooth* dengan *uuid* yang sudah ditentukan.
8. Menambah atribut *bluetooth* berupa *service* yang sudah ditentukan.
9. Menambah atribut *bluetooth* berupa *characteristic* yang sudah ditentukan.
10. Memberikan nilai awal pada status jatuh
11. Membuka *port bluetooth low energy*.

1	<code>inititalize I2C bus</code>
2	<code>serial.begin</code>

```

3  set accelgyro1 to address 0x68
4  set accelgyro2 to address 0x69
5  initialize accelgyro1
6  initialize accelgyro2
7  accelgyro1 test connection
8  accelgyro2 test connection
9  blePheriperal.setLocalName("FallDetection101")
10 blePheriperal.setAdvertisedServiceUuid(fallDet
    ection.uuid())
11 blePheriperal.addAttribute(fallDetection)
12 blePheriperal.addAttribute(fallStatus)
13 blePheriperal.setValue(0)
14 blePheriperal.begin

```

Kode Sumber 4.1 Inisialisasi pada *microcontroller arduino*

Pada proses inisialisasi *microcontroller arduino*, digunakan variabel *accelgyro* untuk mendefinisikan modul GY – 52. Pada sistem ini digunakan dua modul GY – 52, maka pada modul GY – 52 yang pertama diatur ke alamat I2C 0x68 dan modul GY – 52 diatur ke alamat I2C 0x69 agar dapat digunakan secara paralel.

4.3.2 Implementasi Pembacaan Sensor

Pada bagian ini dijelaskan mengenai proses pembacaan data yang didapatkan dari modul GY – 52. Proses pembacaan data sensor memiliki tahapan sebagai berikut:

1. Membaca 6 – axis raw data yang dihasilkan dari modul GY – 52.
2. Pembacaan data pada sensor 1 dilakukan jika pengguna terdeteksi berdiri saja, sedangkan pembacaan data pada sensor 2 dilakukan terus – menerus selama sensor 1 tidak mendeteksi jatuh.
3. Nilai yang didapatkan dari fungsi *getmotion6* masih berupa nilai digital, sehingga perlu dikalibrasi lagi agar nilai tersebut yang didapatkan memiliki satuan yang baku sehingga mempermudah pengolahan data berikutnya. Proses kalibrasi nilai – nilai tersebut dilakukan dengan cara

membagi dengan koefisien sensitivitas sensor. Nilai dari sensor *accelerometer* dibagi dengan koefisien 16384 untuk mendapatkan *range* data ± 2 dengan satuan *g*, sedangkan nilai dari sensor *gyroscope* dibagi dengan 131 untuk mendapatkan *range* data ± 250 dengan satuan $^{\circ}/s$.

4. Hasil nilai dari masing – masing sensor yang sudah terkalibrasi kemudian ditampilkan ke dalam layar.

1	if berdiriValue >= 2
	accelgyro1 getmotion6 (&aX1, &aY1, &aZ1, &gX1,
	&gY1, &gZ1)
3	set accelX1 to aX1/16384
4	set accelY1 to aY1/16384
5	set accelZ1 to aZ1/16384
6	set gyroX1 to gX1/131
7	set gyroY1 to gY1/131
8	set gyroZ1 to gZ1/131
9	print (accelX1, accelY1, accelZ1, gyroX1,
	gyroY1, gyroZ1)
10	if jatuhValue != 2
11	accelgyro2 getmotion6 (&aX2, &aY2, &aZ2, &gX2,
	&gY2, &gZ2)
12	set accelX2 to aX2/16384
13	set accelY2 to aY2/16384
14	set accelZ2 to aZ2/16384
15	set gyroX2 to gX2/131
16	set gyroY2 to gY2/131
17	set gyroZ2 to gZ2/131
18	print (accelX2, accelY2, accelZ2, gyroX2,
	gyroY2, gyroZ2)

Kode Sumber 4.2 Implementasi pembacaan sensor

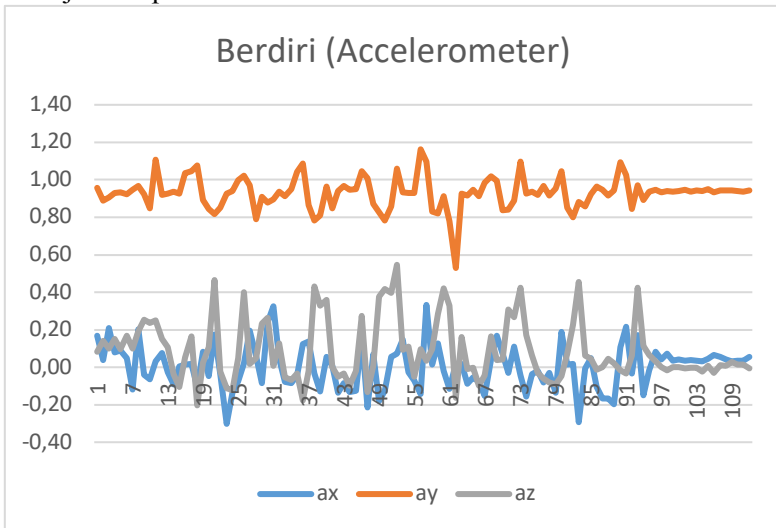
4.3.3 Implementasi Pengambilan Data *Training*

Nilai *threshold* yang digunakan sebagai acuan untuk membandingkan antara jatuh dan aktivitas sehari – hari didapatkan dengan cara pengambilan data *training*. Pengambilan data *training* dilakukan oleh sukarelawan sebagai pengguna yang akan memperagakan gerakan jatuh maupun kegiatan sehari – hari seperti

berdiri, berjalan, merebahkan badan dan duduk. Berikut adalah data *training* yang diperoleh dari setiap aktivitas yang dilakukan.

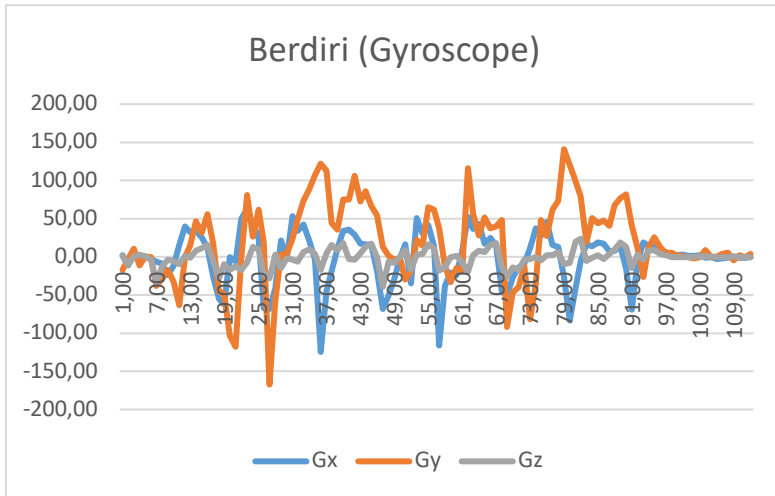
1. Berdiri atau berjalan

Aktivitas ini dilakukan dengan tahap awal pengguna berdiri sejenak kemudian mulai berjalan normal dan diakhiri dengan kembali berdiri. Sensor yang digunakan untuk mengambil data pada saat posisi berdiri adalah sensor 2. Data *training* yang diperoleh dari sensor *accelerometer* maupun sensor *gyroscope* ditunjukkan pada **Gambar 4.2** **Gambar 4.3** **Gambar 4.4**.



Gambar 4.2 Grafik data *training accelerometer* pada aktivitas berdiri dan berjalan

Pada saat pengguna melakukan aktivitas berdiri dengan keadaan diam, grafik data *accelerometer* (**Gambar 4.2**) menunjukkan aktivitas yang rendah namun tetap stabil. Sedangkan ketika pengguna mulai berjalan, aktivitas grafik mulai meningkat, namun juga tetap stabil.



Gambar 4.3 Grafik data *training gyroscope* pada aktivitas berdiri dan berjalan

Grafik data *gyroscope* (**Gambar 4.3**) menunjukkan nilai yang sama dengan nilai *accelerometer* walaupun nilainya sangat berbeda jauh, namun pada kondisi berdiri dengan keadaan diam tetap menunjukkan nilai yang rendah dan ketika berjalan nilainya mengalami peningkatan secara stabil.

Grafik selisih resultan kecepatan sudut (**Gambar 4.4**) menunjukkan perubahan yang stabil dari keadaan pengguna yang awalnya diam, kemudian mulai berjalan dan kembali dalam posisi diam.

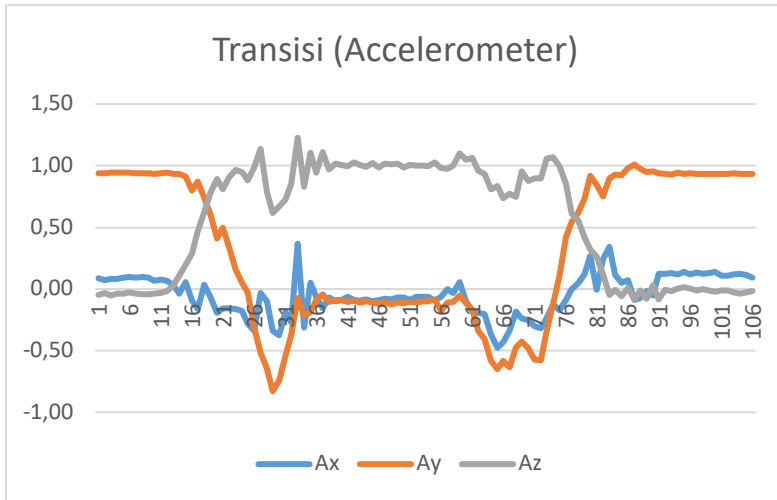


Gambar 4.4 Grafik resultan kecepatan sudut ($\Delta\omega$)

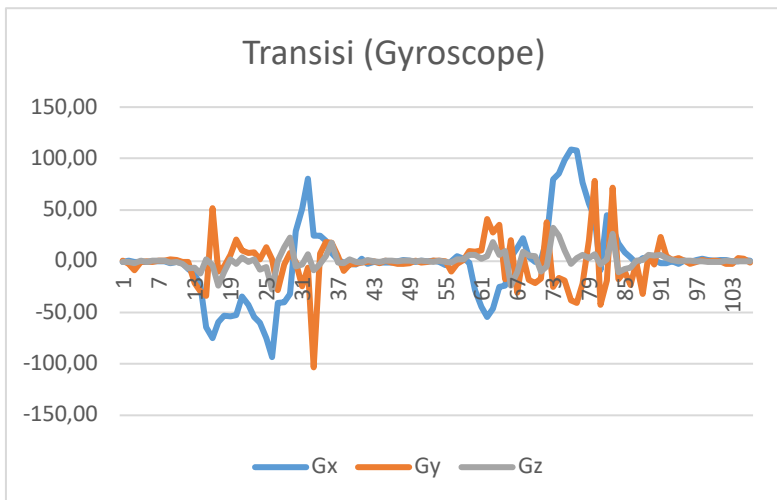
2. Transisi dari posisi berdiri ke posisi duduk

Pada aktivitas ini pengguna memulai dengan posisi berdiri dalam keadaan diam dan kemudian dilanjutkan dengan posisi duduk dan selanjutnya kembali ke posisi berdiri seperti semula. Sensor yang digunakan tetap menggunakan sensor 2. Data *training* yang telah diambil ditunjukkan pada **Gambar 4.5** untuk sensor *accelerometer*, **Gambar 4.6** untuk sensor *gyroscope* dan **Gambar 4.7** untuk nilai resultan kecepatan sudut.

Grafik data *accelerometer* (**Gambar 4.5**) menunjukkan perubahan yang menonjol pada nilai ay dan az ketika pengguna berubah posisi dari yang awalnya berdiri dengan keadaan diam, kemudian duduk dan kembali dalam keadaan berdiri. Dari nilai tersebut dapat ditentukan nilai *threshold* pada *accelerometer* untuk keadaan duduk adalah nilai sumbu $y < 0$ dan nilai dari sumbu $z > 0,8$.



Gambar 4.5 Grafik data *training accelerometer* pada aktivitas transisi dari posisi berdiri ke posisi duduk



Gambar 4.6 Grafik data *training gyroscope* pada aktivitas transisi dari posisi berdiri ke posisi duduk

Grafik data *gyroscope* (**Gambar 4.6**) hanya menunjukkan perbedaan nilai pada saat keadaan transisi saja dari pengguna yang sedang berdiri kemudian duduk dan kembali berdiri. Nilai *gyroscope* pada saat pengguna dalam keadaan berdiri dan duduk tetap menunjukkan nilai yang sama.



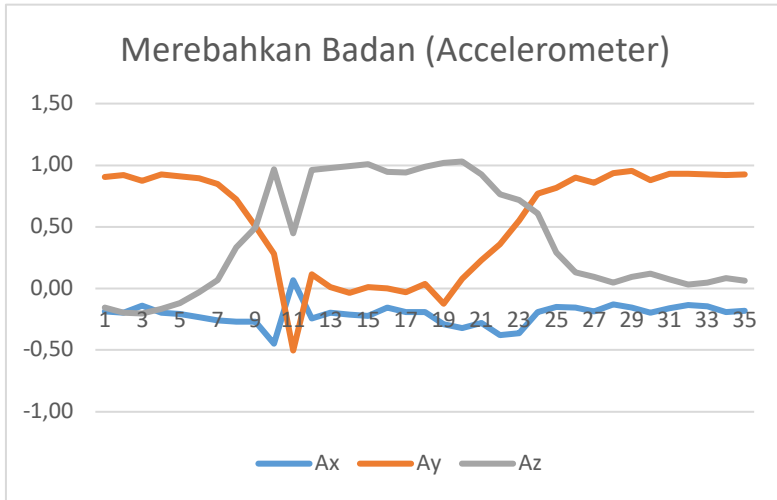
Gambar 4.7 Grafik resultan kecepatan sudut ($\Delta\omega$)

Grafik resultan kecepatan sudut (**Gambar 4.7**) juga menunjukkan hal yang sama dengan nilai *gyroscope*, hanya menunjukkan perbedaan ketika pengguna sedang dalam keadaan transisi. Selebihnya nilai resultan kecepatan sudut menunjukkan kestabilan.

3. Transisi dari posisi duduk ke posisi merebahkan badan

Pada aktivitas ini pengguna mengawali aktivitas dengan posisi duduk dan dilanjutkan dengan melakukan aktivitas merebahkan badan. Pada aktivitas ini digunakan sensor 1 untuk mendapatkan data *training*. Data *training* aktivitas ini ditunjukkan

pada **Gambar 4.8**, **Gambar 4.9**, **Gambar 4.10**, **Gambar 4.11**, dan **Gambar 4.12**.

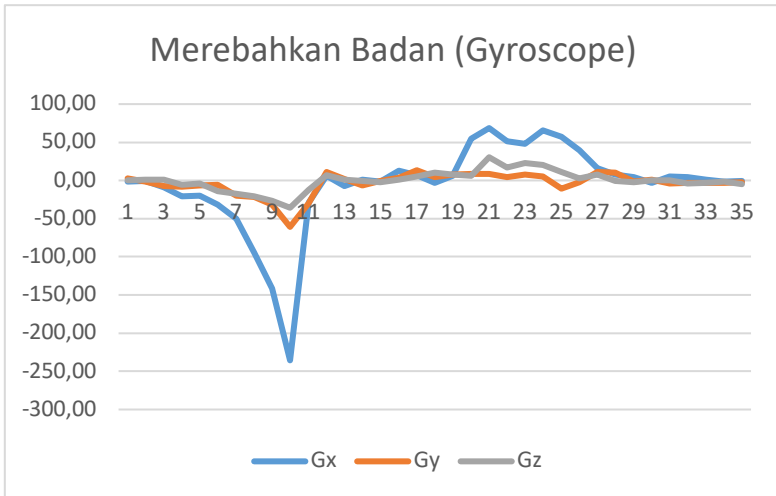


Gambar 4.8 Grafik data *training accelerometer* pada aktivitas merebahkan badan

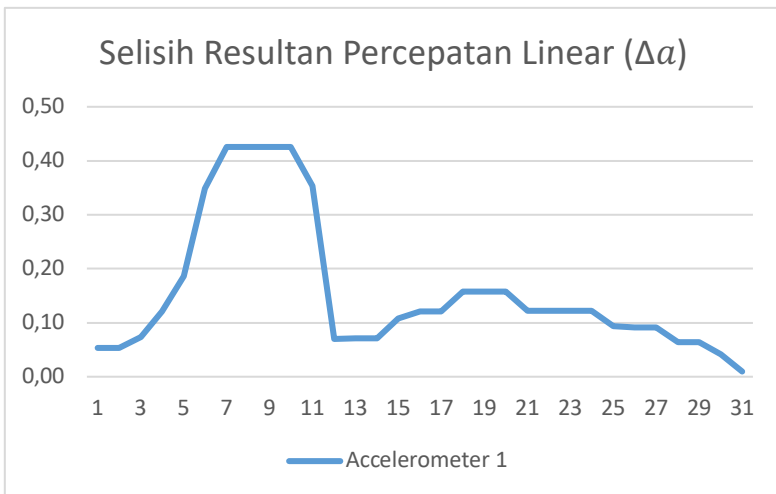
Grafik data *accelerometer* pada saat pengguna memperagakan gerakan merebahkan badan (**Gambar 4.8**) menunjukkan bahwa data pada sumbu y dan z yang mengalami perubahan. Sumbu y menunjukkan penurunan nilai pada saat pengguna merebahkan badan, sedangkan sumbu z mengalami kenaikan.

Pada grafik data *gyroscope* (**Gambar 4.9**) menunjukkan nilai yang relatif memiliki kesamaan dengan data *gyroscope* saat pengguna terdeteksi jatuh, sehingga aktivitas ini yang mungkin menjadi masalah ketika dilakukan pengujian.

Dari hasil data *accelerometer* (**Gambar 4.8**) dan *gyroscope* (**Gambar 4.9**) menunjukkan kemiripan dengan gerakan jatuh, maka dilakukan kembali perhitungan untuk mendapatkan nilai resultan percepatan linear, resultan kecepatan sudut dan perubahan sudut untuk dapat membedakan aktivitas ini dengan gerakan jatuh.

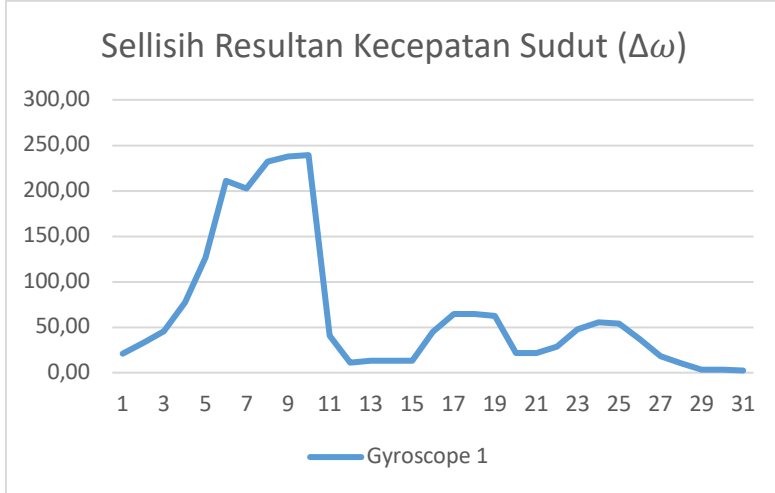


Gambar 4.9 Grafik data *training gyroscope* pada aktivitas merebahkan badan



Gambar 4.10 Grafik selisih resultan percepatan linear (Δa)

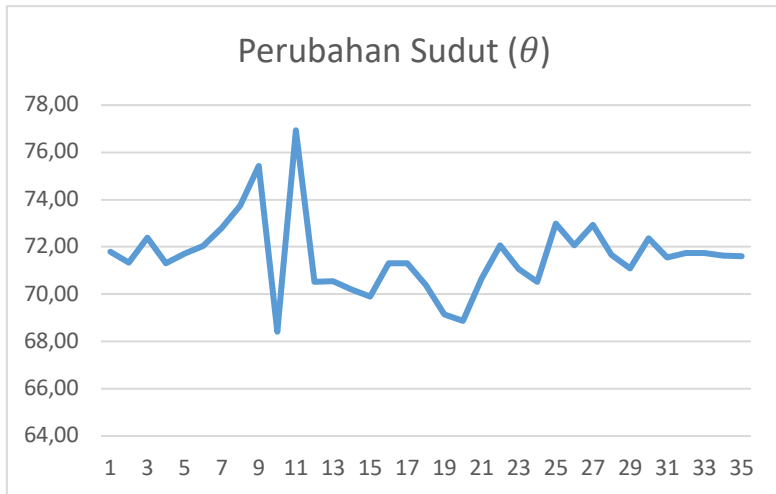
Grafik resultan percepatan linear (**Gambar 4.10**) memperlihatkan perubahan data yang menonjol pada saat posisi pengguna sedang merebahkan badan. Jika dibandingkan dengan grafik data *accelerometer*, grafik ini lebih dapat dibedakan antara posisi sebelum merebahkan badan dan sesudah merebahkan badan.



Gambar 4.11 Grafik selisih resultan kecepatan sudut ($\Delta\omega$)

Grafik resultan kecepatan sudut untuk gerakan merebahkan badan (**Gambar 4.11**) memiliki perbedaan yang signifikan dengan gerakan pada aktivitas lain seperti pada saat transisi dari berdiri ke duduk. Tetapi memiliki kemiripan nilai dengan gerakan jatuh.

Karena gerakan merebahkan badan memiliki sedikit kemiripan dengan gerakan jatuh, digunakan nilai perubahan sudut (**Gambar 4.12**) untuk membedakan gerakan ini dengan gerakan jatuh.

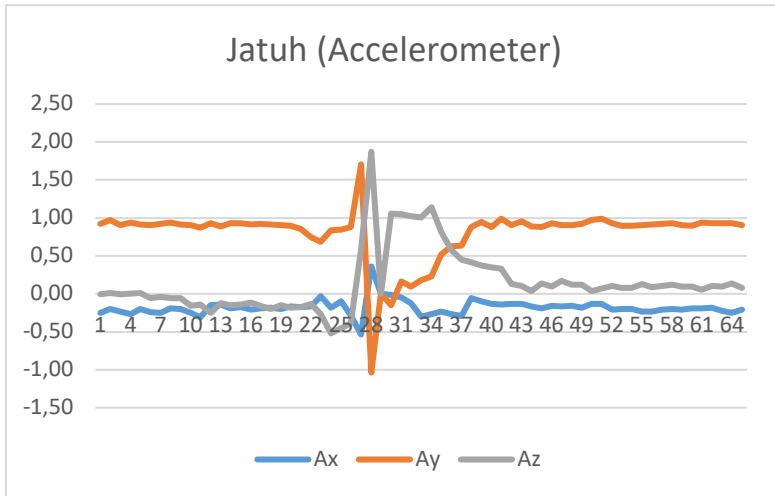


Gambar 4.12 Grafik perubahan sudut (θ)

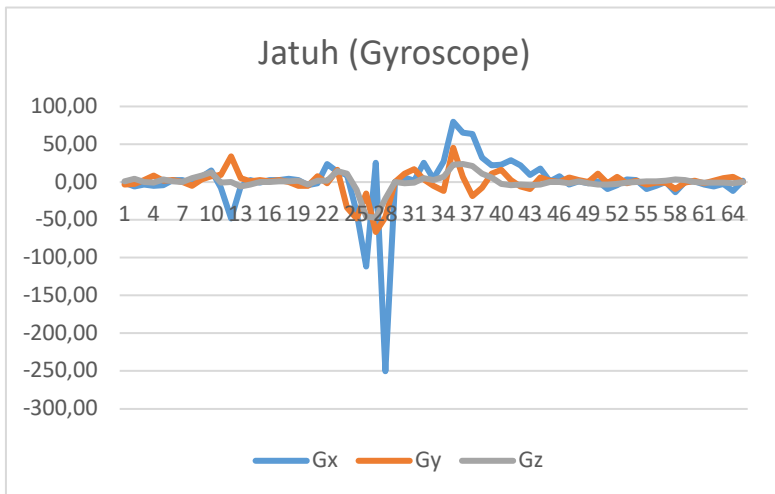
4. Gerakan jatuh

Gerakan jatuh dimulai dengan posisi berdiri dalam keadaan diam selanjutnya dilakukan gerakan jatuh ke belakang. Pengambilan data gerakan jatuh ini dilakukan untuk mendapatkan *threshold* jatuh yang akan digunakan untuk membedakan dengan aktivitas atau gerakan selain jatuh. Data gerakan jatuh ini menggunakan sensor 1 untuk proses pengambilannya. Data *training* untuk gerakan ini dapat dilihat pada **Gambar 4.13**, **Error! Reference source not found.**, **Gambar 4.15**, **Gambar 4.16**, dan **Gambar 4.17**.

Grafik data *accelerometer* pada gerakan jatuh (**Gambar 4.13**) menunjukkan bahwa ada penurunan nilai pada sumbu y dan kenaikan nilai pada sumbu z yang drastis.

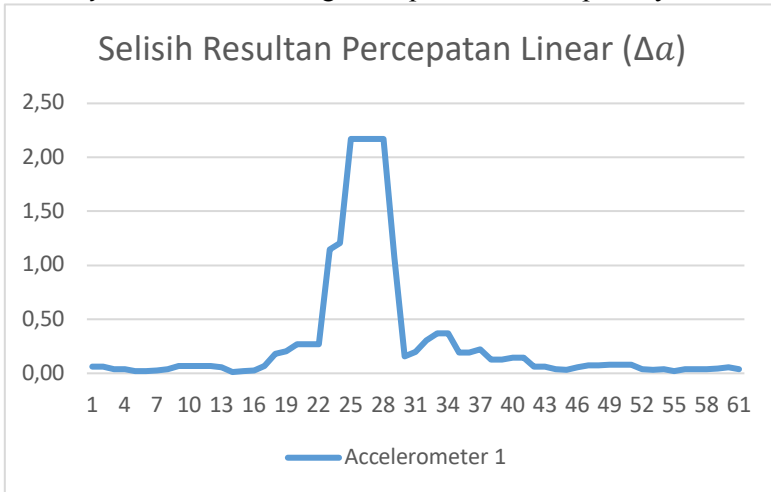


Gambar 4.13 Grafik data *training accelerometer* pada gerakan jatuh



Gambar 4.14 Grafik data *training gyroscope* pada gerakan jatuh

Pada grafik data *gyroscope* (**Gambar 4.14**) dapat dilihat bahwa nilai pada sumbu x mengalami penurunan drastis pada saat pengguna memperagakan gerakan jatuh. Sedangkan nilai pada sumbu y dan sumbu z mengalami penurunan tetapi hanya sedikit.

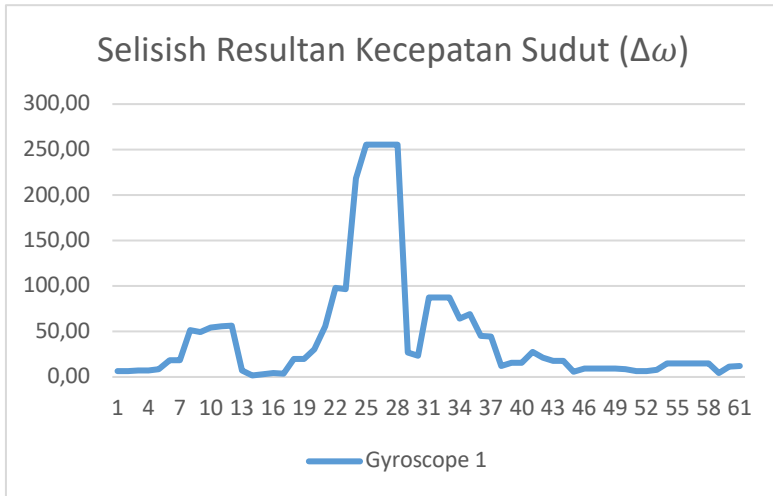


Gambar 4.15 Grafik selisih resultan percepatan linear (Δa)

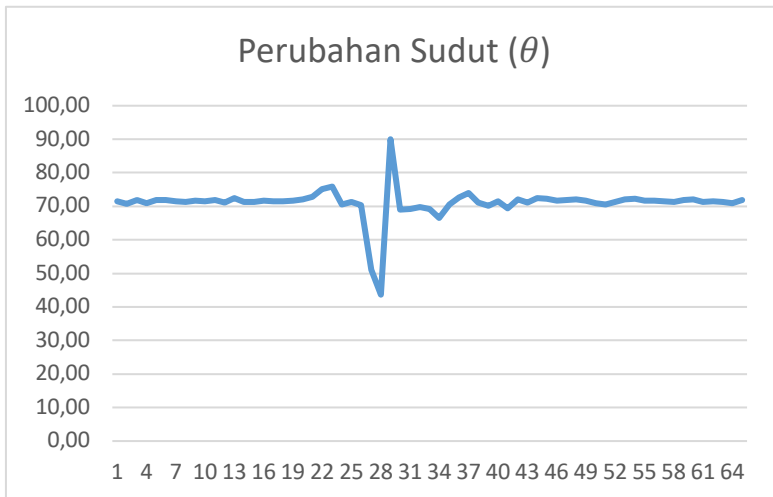
Grafik resultan percepatan linear (**Gambar 4.15**) menunjukkan perubahan nilai yang lebih jelas pada saat pengguna belum jatuh dan pada saat pengguna memperagakan gerakan jatuh. Dari data tersebut dapat ditentukan nilai *threshold* Δa untuk menentukan seseorang jatuh adalah 1,00.

Nilai pada grafik resultan kecepatan sudut (**Gambar 4.16**) mengalami perubahan yang sangat drastis ketika pengguna sedang memperagakan gerakan jatuh. Dari nilai tersebut dapat ditentukan nilai *threshold* $\Delta \omega$ untuk menentukan seseorang jatuh adalah 200.

Nilai perubahan sudut (**Gambar 4.17**) digunakan untuk memperjelas perbedaan antara gerakan pada aktivitas merebahkan badan dan gerakan jatuh. Dari data *training* yang sudah diambil, dapat ditentukan nilai *threshold* θ untuk menentukan seseorang jatuh adalah 60.



Gambar 4.16 Grafik selisih resultan kecepatan sudut ($\Delta\omega$)



Gambar 4.17 Grafik perubahan sudut (θ)

4.3.4 Implementasi Algoritma *Fall Detection* Menggunakan Metode *Quad Threshold*

Implementasi algoritma *fall detection* menggunakan metode *quad threshold* diawali dengan pengambilan data *training*. Pengambilan data *training* bertujuan untuk mendapatkan data yang nantinya akan digunakan sebagai acuan untuk menentukan *threshold aktivitas* berdiri dan aktivitas jatuh. Setelah *threshold* didapatkan, selanjutnya dilakukan implementasi pada perangkat lunak untuk melakukan perbandingan data *training* terhadap data pengguna yang sedang melakukan kegiatan sehari – hari. Tahapan proses deteksi jatuh menggunakan algoritma *fall detection* dengan metode *quad threshold* adalah sebagai berikut:

1. Menghitung nilai resultan percepatan linear dan nilai kecepatan sudut pada sensor 1
2. Menghitung nilai perubahan sudut pada sensor 1
3. Mencari nilai maksimum dan minimum dari nilai resultan percepatan linear dan nilai kecepatan sudut pada sensor 1 dengan interval 5 nilai
4. Menghitung nilai selisih antara nilai maksimum dan minimum tersebut sesuai intervalnya.
5. Jika nilai selisih percepatan linear, selisih kecepatan sudut, dan perubahan sudut pada sensor 1 melewati *threshold* jatuh yang sudah ditentukan maka pengguna terdeteksi jatuh.
6. Ketika pengguna terdeteksi jatuh, *microcontroller arduino* akan mengirim status jatuh ke perangkat android melalui *bluetooth*.

Sedangkan proses deteksi posisi berdiri menggunakan algoritma *fall detection* dengan metode *quad threshold* adalah sebagai berikut:

1. Menghitung nilai resultan kecepatan sudut pada sensor 2.

2. Mencari nilai maksimum dan minimum dari nilai resultan kecepatan sudut pada sensor 2 dengan interval 5 nilai.
3. Menghitung nilai selisih antara nilai maksimum dan minimum tersebut sesuai intervalnya.
4. Jika nilai *accelerometer* pada sumbu y dan sumbu z dan nilai kecepatan sudut pada sensor 2 melewati threshold jatuh yang sudah ditentukan maka pengguna terdeteksi duduk (tidak berdiri).
5. Ketika pengguna tidak terdeteksi berdiri maka sensor 1 akan dimatikan dan tidak akan membaca data.

```

1  set M_PI to 3.14159265358979323846
2  set resultAccel1 to sqrt((pow(accelX1, 2))
   + (pow(accelY1, 2)) + (pow(accelZ1, 2)))
3  set resultGyrol to sqrt((pow(gyroX1, 2))
   + (pow(gyroY1, 2)) + (pow(gyroZ1, 2)))
4  set tetha to (acos(resultAccel)) * (180.00/M_PI)
5  if count == 4
6    set count to count - 1
7    for i = 0 to count
8      set tempAccel1[i] to tempAccel1[i + 1]
9      set tempGyrol[i] to tempGyrol[i + 1]
10  if count < 4
11    set tempAccel1[count] to resultAccel1
12    set tempGyrol[count] to resultGyrol
13  set count to count + 1
14  for i = 0 to count
15    if tempAccel1[i] > maxAccel1
16      set maxAccel1 to tempAccel1[i]
17    else if tempAccel1[i] < minAccel1
18      set minAccel1 to tempAccel1[i]
19    if tempGyrol[i] > maxGyrol
20      set maxGyrol to tempGyrol[i]
21    else if tempGyrol[i] < minGyrol
22      set minGyrol to tempGyrol[i]
23  if count == 4
24    set deltaAccel1 to maxAccel1 - minAccel1
25    set deltaGyrol to maxGyrol - minGyrol
26    set maxAccel1 to -99
27    set minAccel1 to 99
28    set maxGyrol to -9999

```

```

29      set minGyro1 to 9999
30      if deltaAccel1 != null && deltaGyro1 != null
31          if deltaAccel1 > thresholdAccel1
32              && deltaGyro1 > thresholdGyro1
33              && tetha < thresholdTetha
34                  set jatuhValue to 2
35                  fallStatus.setValue(jatuhValue)
36          else
37              set jatuhValue to 0
38              fallStatus.setValue(jatuhValue)

```

Kode Sumber 4.3 Implementasi algoritma *fall detection* untuk sensor 1

```

1      set resultGyro2 to sqrt((pow(gyroX2, 2))
2      +(pow(gyroY2, 2))+(pow(gyroZ2,2)))
3      if count == 4
4          set count to count - 1
5          for i = 0 to count
6              set tempGyro2[i] to tempGyro2[i + 1]
7          if count < 4
8              set tempGyro2[count] to resultGyro2
9          set count to count + 1
10         for i = 0 to count
11             if tempGyro2[i] > maxGyro2
12                 set maxGyro2 to tempGyro2[i]
13             else if tempGyro2[i] < minGyro2
14                 set minGyro2 to tempGyro2[i]
15         if count == 4
16             set deltaGyro2 to maxGyro2 - minGyro2
17             set maxGyro1 to -9999
18             set minGyro1 to 9999
19         if deltaGyro2 != null
20             if deltaGyro > thresholdGyro2
21                 set berdirValue to berdirValue + 1
22             if accelY2 < thresholdAccelY2 && accelZ2 >
23                 thresholdAccelZ2
24                 set berdirValue to berdirValue + 1
25             else set berdirValue to 0

```

Kode Sumber 4.4 Implementasi algoritma *fall detection* untuk sensor 2

4.4 Implementasi Aplikasi Android *Fall Detection*

Aplikasi *fall detection* dalam sistem ini bertindak sebagai perantara antara pengguna dengan perangkat android yang memudahkan pengguna jika ingin melihat hasil data dari *microcontroller arduino*. Selain itu juga, aplikasi android ini yang mengatur secara otomatis notifikasi yang akan dikirim ke keluarga / kerabat pengguna sesuai dengan hasil deteksi yang didapat oleh *microcontroller arduino*. Penjelasan mengenai implementasi dari aplikasi *fall detection* adalah sebagai berikut:

4.4.1 Implementasi Perizinan Menggunakan *Bluetooth*, Lokasi, dan Mengirimkan SMS pada Perangkat Android

Untuk menjalankan aplikasi *fall detection* pada perangkat android, diperlukan izin menggunakan *bluetooth*, izin menggunakan lokasi, dan izin mengirimkan sms. Izin tersebut diperlukan karena fungsi utama dari aplikasi ini adalah menerima data melalui komunikasi antar perangkat *bluetooth*, menentukan lokasi perangkat saat pengguna terdeteksi jatuh dan mengirimkan notifikasi berupa sms. Kode sumber dari perizinan menggunakan fasilitas tersebut dapat dilihat pada **Kode Sumber 4.5**, **Kode Sumber 4.6**, **Kode Sumber 4.7**, **Kode Sumber 4.8**, dan **Kode Sumber 4.9**.

1	<code>set bluetoothAdapter to bluetoothManager</code>
2	<code>.getAdapter</code>
3	<code>if bluetoothAdapter != null</code>
	<code>return true</code>

Kode Sumber 4.5 Melakukan pengecekan apakah perangkat android sudah memiliki *bluetooth*

1	<code>if getPackageManager</code>
	<code>.hasSystemFeature(PackageManager</code>
	<code>.FEATURE_BLUETOOTH_LE)</code>
2	<code>return true</code>
3	<code>else return false</code>

Kode Sumber 4.6 Melakukan pengecekan apakah perangkat android *support* dengan *bluetooth low energy*

1	<code>if !bluetoothAdapter.isEnabled</code>
2	<code>Intent enableBTIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE)</code>
3	<code>startActivityForResult(enableBTIntent, 101)</code>

Kode Sumber 4.7 Izin untuk menggunakan dan mengaktifkan *bluetooth* jika belum diaktifkan

1	<code>if ActivityCompat.checkSelfPermission</code> <code>(this, Manifest.permission</code> <code>.ACCESS_FINE_LOCATION) != PackageManager</code> <code>.PERMISSION_GRANTED and ActivityCompat</code> <code>.checkSelfPermission(this, Manifest</code> <code>.permission.ACCESS_COARSE_LOCATION)</code> <code>!= PackageManager.PERMISSION_GRANTED</code>
2	<code>ActivityCompat.requestPermissions(this,</code> <code>new String[]{Manifest.permission</code> <code>.ACCESS_FINE_LOCATION}, 101)</code>
3	<code>ActivityCompat.requestPermissions(this,</code> <code>new String[]{Manifest.permission</code> <code>.ACCESS_COARSE_LOCATION}, 101);</code>

Kode Sumber 4.8 Izin untuk menggunakan lokasi perangkat android

1	<code>if ContextCompat.checkSelfPermission</code> <code>(this.getApplicationContext, Manifest</code> <code>.permission.SEND_SMS) != PackageManager</code> <code>.PERMISSION_GRANTED)</code>
2	<code>ActivityCompat.requestPermissions(this,</code> <code>new String[]{Manifest.permission</code> <code>.SEND_SMS}, 101)</code>

Kode Sumber 4.9 izin untuk menggunakan fitur sms pada perangkat android

4.4.2 Implementasi Fitur Menambah, Mengubah dan Menghapus Kontak

Kontak yang berisi nama dan nomor telepon dalam aplikasi *fall detection* sangat berguna dalam berjalannya aplikasi ini. Jika

tidak ada kontak yang tersimpan dalam daftar kontak, aplikasi akan tetap berjalan, akan tetapi notifikasi yang berupa sms tidak akan dikirim kepada siapapun dan aplikasi hanya akan menampilkan pemberitahuan jatuh di perangkat android milik pengguna saja. Untuk itu fitur tambah, ubah, dan hapus kontak harus ada dalam aplikasi *fall detection*.

Proses tambah kontak dimulai ketika pengguna menekan tombol tambah kontak pada halaman awal aplikasi. Selanjutnya pengguna akan mengisi kolom nama dan nomor telepon dan menekan tombol simpan. Setelah tombol simpan ditekan, aplikasi akan menyimpan rincian kontak ke dalam *database* di dalam perangkat android. Jika data rincian kontak diperlukan, maka data dapat diambil dari *database* tersebut.

Proses mengubah kontak memiliki dasar yang sama dengan menambah kontak. Perbedaannya adalah ketika mengubah kontak, diperlukan data yang sudah ada dalam *database* dan selanjutnya data tersebut diubah oleh pengguna. Sedangkan proses menghapus data dilakukan dengan cara mengambil data kontak yang ada dalam *database* kemudian data tersebut dihapus dari *database*. Data kontak yang sudah dihapus dari *database* sudah tidak dapat dikembalikan lagi. Kode sumber dari fitur tambah, ubah, dan hapus kontak dapat dilihat pada **Kode Sumber 4.10**, **Kode Sumber 4.11**, dan **Kode Sumber 4.12**.

1	initialize contact
2	set contactID to (int)System.currentTimeMillis/1000
3	set contactName to name
4	set contactNumber to number
5	realm.beginTransaction
6	realm.copyToRealm(contact)
7	realm.commitTransaction

Kode Sumber 4.10 Menambahkan data kontak ke *database* berdasarkan *inputan* pengguna

1	realm.beginTransaction
2	set contact to realm.where(Contact.class) .equalTo(id).findFirst

3	set contactName (name)
4	set contactNumber (number)
5	realm.commitTransaction

Kode Sumber 4.11 Mengubah data kontak

1	set dataResult to realm.where(Contact.class) .equalTo(id).findAll
2	realm.beginTransaction
3	dataResult.deleteAllFromRealm
4	realm.commitTransaction

Kode Sumber 4.12 Menghapus kontak dari database

4.4.3 Implementasi Fitur Mencari Perangkat *Bluetooth*

Pencarian perangkat *bluetooth* dilakukan secara otomatis setelah pengguna berada pada halaman pencarian *bluetooth*, namun pengguna juga dapat melakukan pencarian secara manual dengan menekan tombol cari. Proses pencarian perangkat *bluetooth* dilakukan selama selang waktu tertentu. Setelah perangkat android menemukan perangkat yang terintegrasi dengan *bluetooth* di sekitarnya, nama – nama perangkat *bluetooth* yang ada kemudian ditampilkan pada halaman pencarian *bluetooth*. Untuk melakukan pencarian terhadap perangkat *bluetooth* digunakan fungsi *startLeScan()*. Fungsi tersebut menggunakan *BluetoothAdapter.LeScanCallback* sebagai parameternya. Pada proses pencarian *bluetooth* diperlukan penggunaan baterai yang besar, untuk itu diperlukan perlakuan khusus dalam fitur ini seperti membatasi waktu proses pencarian perangkat *bluetooth* dan tidak menggunakan fungsi *loop* pada proses pencarian. Kode sumber untuk melakukan pencarian perangkat *bluetooth* dapat dilihat pada **Kode Sumber 4.13 dan Kode Sumber 4.14.**

1	private void scanLeDevice
2	handler.postDelayed(new Runnable)
3	@Override
4	void function run
5	set scanningBluetooth to false
6	bluetoothAdapter
	.stopLeScan (LeScanCallback)
7	set delay to SCAN_PERIOD

8	set scanningBluetooth to true
9	bluetoothAdapter .startLeScan(LeScanCallback)

Kode Sumber 4.13 Fungsi mencari dan mengatur *delay* pencarian perangkat *bluetooth*

1	set LeScanCallback to
2	BluetoothAdapter.LeScanCallback
3	@Override
4	public void onLeScan
5	runOnUiThread(new Runnable())
6	@Override
7	public void run
8	LeDeviceListAdapter
	.addDevice(device)
	LeDeviceListAdapter
	.notifyDataSetChanged

Kode Sumber 4.14 Fungsi menyimpan hasil pencarian perangkat *bluetooth* ke dalam *list*

4.4.4 Implementasi Koneksi Perangkat Android dengan Perangkat *Bluetooth*

Koneksi perangkat android dengan perangkat *bluetooth* dilakukan secara otomatis setelah pengguna berada pada halaman *fall detection*. Jika perangkat android tidak bisa terkoneksi dengan perangkat *bluetooth* secara otomatis, pengguna juga dapat melakukan koneksi secara manual dengan menekan tombol *connect* pada halaman *fall detection*. Untuk melakukan interaksi dengan perangkat *bluetooth*, dilakukan inisialisasi terhadap *bluetooth service*. Selanjutnya dilakukan koneksi ke GATT server pada perangkat *bluetooth* dengan menggunakan fungsi *connectGatt()*. Dari fungsi tersebut didapatkan pola *BluetoothGatt instance* untuk mengolah GATT pada perangkat android. Kode sumber koneksi perangkat android dan perangkat *bluetooth* dapat dilihat pada **Kode Sumber 4.15** dan **Kode Sumber 4.16**.

1	set bluetoothLeService to
	((BluetoothLeService.LocalBinder) service)
	.getService

2	if bluetoothLeService.initialize
3	bluetoothLeService.connect(deviceAddress)

Kode Sumber 4.15 Melakukan inisialisasi *bluetooth service* dan memanggil fungsi *connect* pada class *BluetoothLeService* untuk melakukan koneksi perangkat android dengan perangkat *bluetooth*

1	public boolean connect(final String address)
2	if bluetoothAdapter == null or address == null
	return false
3	if bluetoothDeviceAddress != null and address.equals(bluetoothDeviceAddress) and bluetoothGatt != null
4	if bluetoothGatt.connect
5	set connectionState to STATE_CONNECTING
6	return true
7	else return false
8	set device to bluetoothAdapter .getRemoteDevice(address)
9	if device == null
10	return false
11	set bluetoothGatt to device.connectGatt (this, false, gattCallback)
12	set bluetoothDeviceAddress to address
13	set connectionState to STATE_CONNECTING
14	return true

Kode Sumber 4.16 Fungsi *connect* pada class *BluetoothLeDevice*

4.4.5 Implementasi Proses Deteksi Jatuh dan Pengiriman Notifikasi Sms

Setelah perangkat terhubung dengan perangkat *bluetooth*, perangkat android akan meminta *microcontroller arduino* untuk mengirimkan data yang sudah diolah. *Microcontroller arduino* hanya akan mengirimkan data berupa status bahwa pengguna jatuh atau tidak. Setelah pengguna terdeteksi jatuh, *microcontroller arduino* akan mengirimkan status jatuh ke perangkat android

melalui komunikasi *bluetooth*. Data status jatuh yang diterima perangkat android kemudian diproses untuk selanjutnya perangkat android melalui aplikasi *fall detection* mengirimkan notifikasi berupa sms kepada semua kontak yang sudah terdaftar pada aplikasi *fall detection*. Proses deteksi dan pengiriman notifikasi sms dapat dilihat pada **Kode Sumber 4.17** dan **Kode Sumber 4.18**.

1 2 3	<pre>@Override protected void onResume() { super.onResume(); registerReceiver(gattUpdateReceiver, makeGattUpdateIntentFilter()); }</pre>
-------------	--

Kode Sumber 4.17 Fungsi untuk mendapatkan data secara terus menerus dari perangkat *bluetooth*

1 2 3 4 5 6 7 8 9	<pre>set gattUpdateReceiver to new BroadcastReceiver() @Override public void onReceive(Context context, Intent intent) { if (BluetoothLeService .ACTION_DATA_AVAILABLE.equals(action)) { set status to intent.getStringExtra(BluetoothLeService.EXTRA_DATA) displayData(status) } if (status.equals(fall)) { set listContacts to realmHelper .findAllContact() sentSms(listContacts, messages) } }</pre>
---	---

Kode Sumber 4.18 Fungsi *BroadcastReceiver* untuk menampilkan data pada halaman *fall detection* dan mengirimkan notifikasi sms kepada semua kontak yang tersimpan jika pengguna terdeteksi jatuh

BAB V

HASIL UJI COBA DAN EVALUASI

Bab ini berisi penjelasan mengenai uji coba dengan analisa yang telah dilakukan beserta skenario uji coba dan evaluasi. Hasil uji coba didapatkan dari implementasi pada bab sebelumnya dengan skenario yang berbeda. Bab ini berisikan pembahasan mengenai lingkungan pengujian, data pengujian, dan uji kinerja.

5.1 Lingkungan Uji Coba

Lingkungan pengujian pada uji coba menjelaskan tentang lingkungan dimana sistem akan diuji coba. Uji coba ini menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada **Tabel 5.1**.

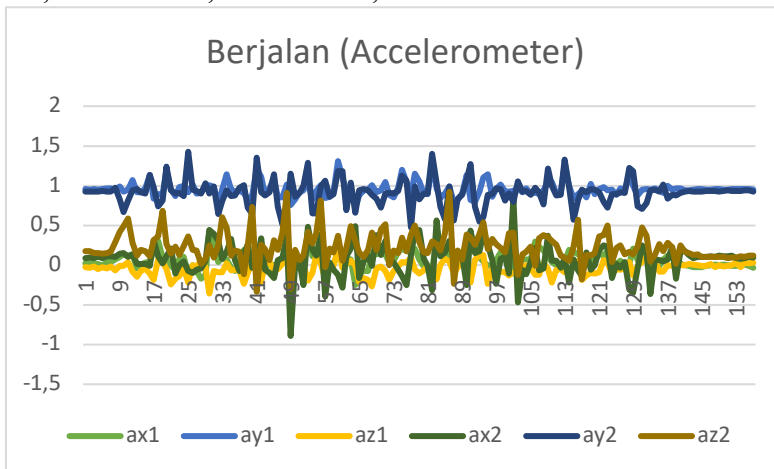
Tabel 5.1 Lingkungan pengujian

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Laptop MacBook Pro 13 – inch	Intel® Core i5 2.7 GHz, 8 Gb 1867 MHz DDR3 RAM
	Arduino 101	Intel® Curie 32 MHz (7 – 12V), 196 Kb Flash Memory, 24 Kb SRAM
	Smartphone Samsung Galaxy S7 Flat	Samsung Exynos Octa 8890 2.60 GHz, 4 Gb RAM

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Lunak	Sistem Operasi Perangkat Android	Android 7.0 Nougat

5.2 Data Uji Coba

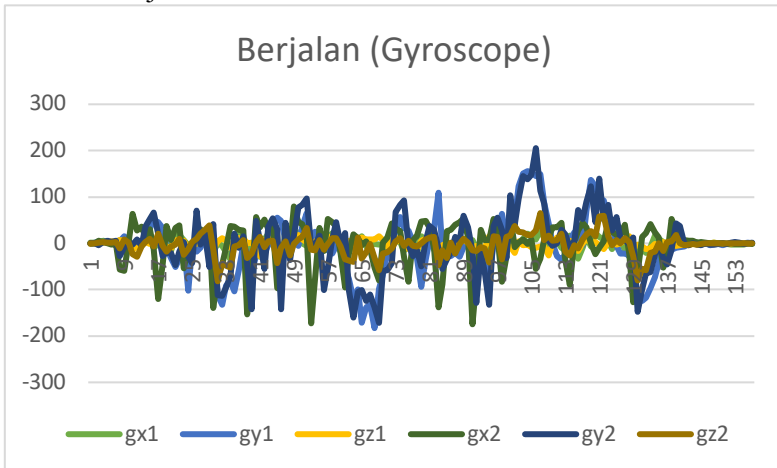
Data uji coba yang digunakan sebagai masukan adalah data yang diambil dari sukarelawan yang memperagakan gerak jatuh maupun gerakan dalam aktivitas sehari – hari. Percobaan memperagakan gerakan jatuh dan gerakan melakukan aktivitas sehari – hari seperti duduk, merebahkan badan dan berjalan dilakukan sebanyak 60 kali. Contoh data dapat dilihat pada **Gambar 5.1, Gambar 5.2, Gambar 5.3, Gambar 5.4, Gambar 5.5, Gambar 5.6, Gambar 5.7, dan Gambar 5.8.**



Gambar 5.1 Grafik data *accelerometer* uji coba berjalan

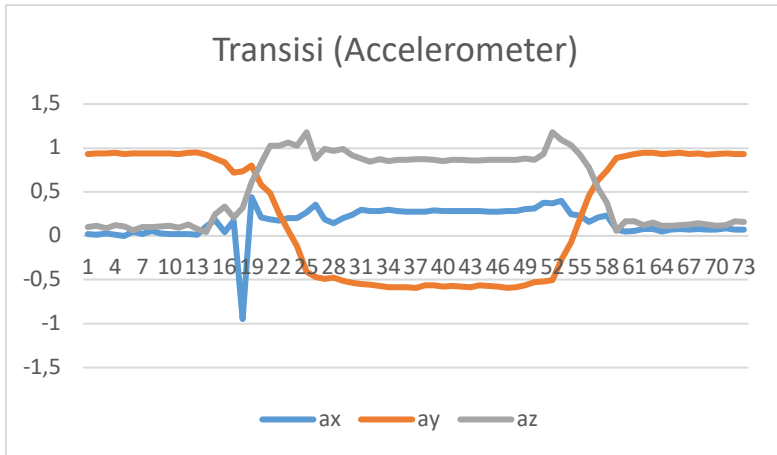
Grafik data *accelerometer* pada saat pengguna memperagakan aktivitas berjalan (**Gambar 5.1**) menunjukkan nilai grafik yang stabil, tidak ada nilai yang naik terlalu drastis ataupun turun terlalu drastis.

Grafik data *gyroscope* (**Gambar 5.2**) juga memperlihatkan nilai grafik yang stabil pada saat pengguna memperagakan aktivitas berjalan.

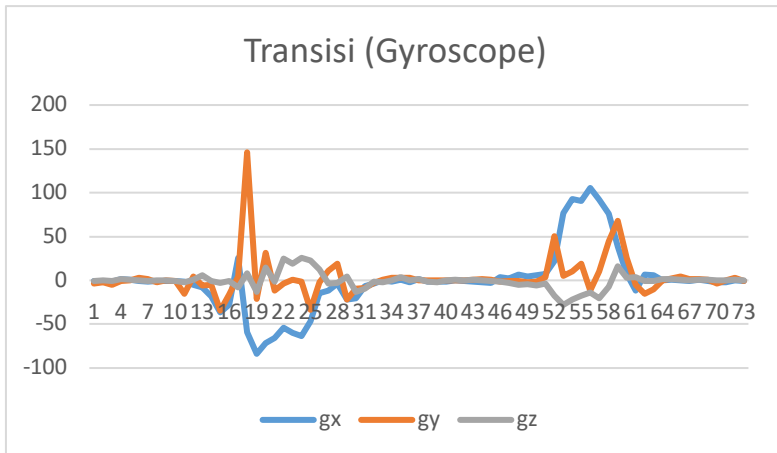


Gambar 5.2 Grafik data *gyroscope* uji coba berjalan

Grafik data *accelerometer* (**Gambar 5.3**) menunjukkan penurunan nilai pada sumbu y dan kenaikan nilai pada sumbu z yang dimulai pada data ke 19. Nilai *accelerometer* pada sumbu y telah melewati batas *threshold* duduk pada sumbu y yaitu $< 0,0$ ketika pengguna berubah posisinya dari yang sebelumnya berdiri kemudian duduk. Begitu juga nilai *accelerometer* pada sumbu z juga telah melewati batas *threshold* duduk pada sumbu z yaitu $> 0,8$. Dengan begitu, dapat diketahui bahwa pengguna sedang duduk. Kemudian pada data ke 52 mulai menunjukkan kenaikan pada sumbu y dan penurunan pada sumbu z yang menunjukkan bahwa pengguna kembali berdiri.

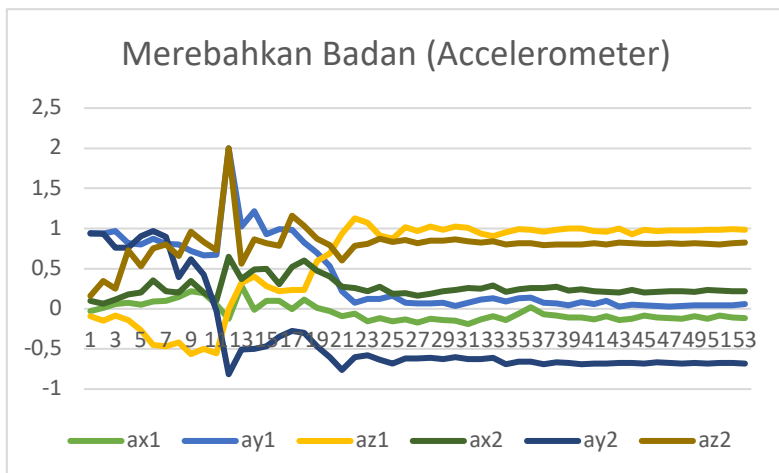


Gambar 5.3 Grafik data *accelerometer* uji coba transisi

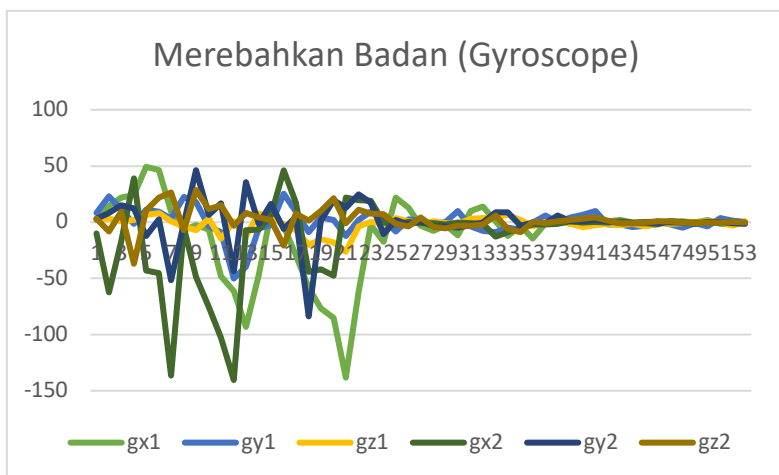


Gambar 5.4 Grafik data *gyroscope* uji coba transisi

Grafik data *gyroscope* (**Gambar 5.4**) hanya menunjukkan perubahan nilai ketika pengguna mengubah posisinya dari posisi berdiri ke duduk maupun sebaliknya, selebihnya nilai tetap stabil.

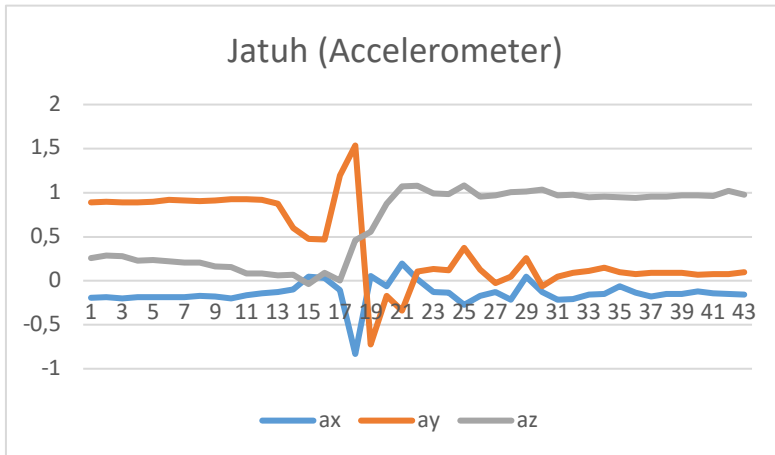


Gambar 5.5 Grafik data *accelerometer* uji coba merebahkan badan



Gambar 5.6 Grafik data *gyroscope* uji coba merebahkan badan

Grafik data *accelerometer* (**Gambar 5.5**) mulai menunjukkan perubahan nilai pada data ke 9 saat pengguna memperagakan gerakan merebahkan badan. Sedangkan pada grafik data *gyroscope* (**Gambar 5.6**) menunjukkan lebih banyak perubahan nilai pada saat pengguna memperagakan gerakan merebahkan badan.

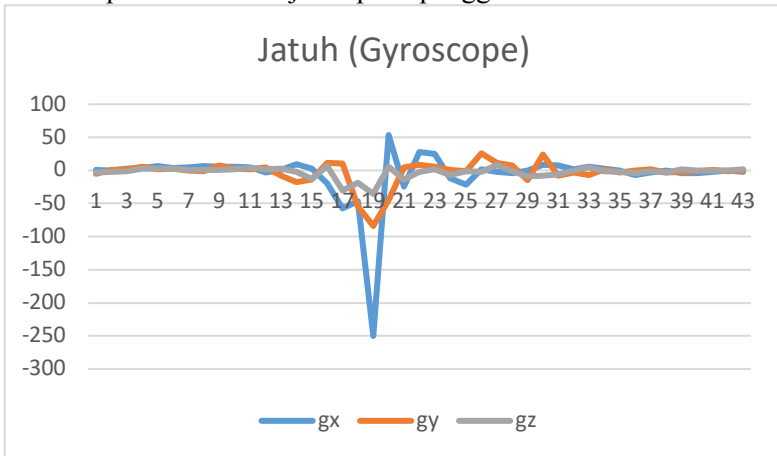


Gambar 5.7 Grafik data *accelerometer* uji coba jatuh

Grafik data jatuh *accelerometer* (**Gambar 5.7**) menunjukkan perubahan nilai pada saat pengguna sedang memperagakan gerakan jatuh yang dimulai pada data ke 15. Sedangkan grafik data *gyroscope* (**Gambar 5.8**) menunjukkan perubahan nilai yang besar terutama pada sumbu y pada saat pengguna memperagakan gerakan jatuh yang juga dimulai pada data ke 15.

Data yang ditampilkan sebagai contoh tersebut belum dilakukan perhitungan untuk mencari Δa , $\Delta \omega$, dan θ . Sehingga hanya dapat diketahui perubahan – perubahan nilainya saja dan belum dapat digunakan untuk mendeteksi jatuh. Sedangkan pada uji coba performa sistem dan analisis hasil uji coba, data yang

digunakan sudah diolah dan didapatkan nilai – nilai Δa , $\Delta \omega$, dan θ untuk dapat mendeteksi jatuh pada pengguna.



Gambar 5.8 Grafik data *gyroscope* uji coba jatuh

5.3 Skenario Uji Coba

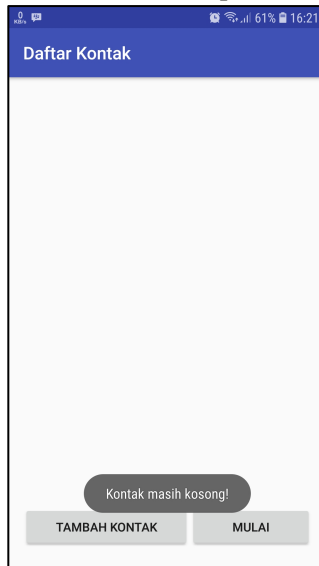
Sebelum dilakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario ini, perangkat akan diuji apakah dapat berjalan dengan benar dan bagaimana performa dari sistem yang telah diimplementasikan. Skenario uji coba ini dibagi menjadi dua bagian utama, yaitu uji coba performa sistem dan uji coba fungsionalitas.

5.3.1 Uji Coba Fungsionalitas

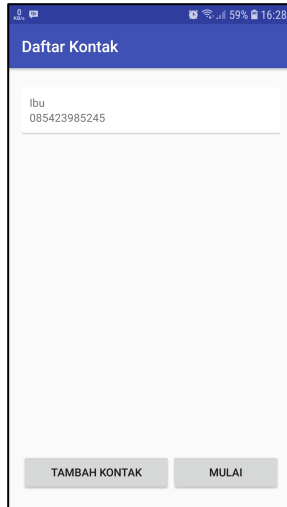
Pada uji coba fungsionalitas akan dijelaskan mengenai pengujian terhadap fungsionalitas aplikasi yang telah dibuat. Hasil uji coba dilakukan pada perangkat android Samsung Galaxy S7 Flat dan ditunjukkan dengan hasil tangkapan layar.

1. Uji coba menambahkan kontak

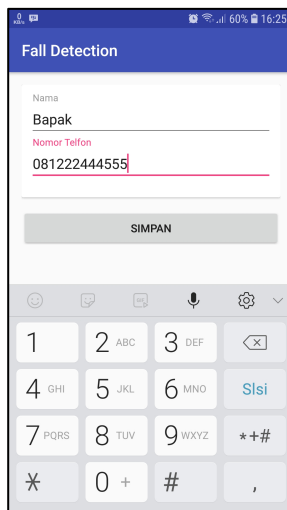
Halaman pertama yang ditampilkan pada aplikasi ini menampilkan daftar kontak yang telah didaftarkan oleh pengguna yang nantinya akan dikirim notifikasi berupa sms jika pengguna terdeteksi jatuh. Aplikasi akan tetap berjalan walaupun kontak masih kosong. Untuk menambahkan kontak pada aplikasi ini, pengguna dapat menekan tombol tambah kontak, kemudian pengguna akan diarahkan ke halaman tambah kontak yang berisi *form* pengisian untuk mengisi nama dan nomor telepon yang akan ditambahkan. Setelah mengisi *form*, pengguna dapat menekan tombol simpan untuk menyimpan data kontak dan pengguna akan kembali diarahkan ke halaman awal aplikasi.



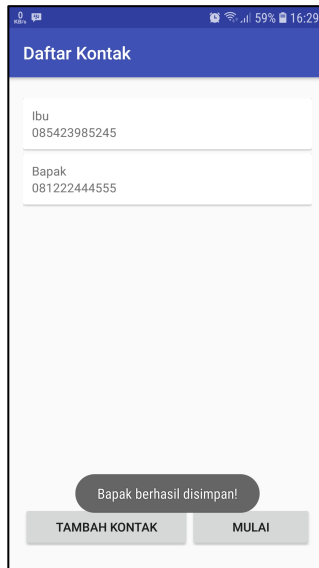
Gambar 5.9 Halaman awal aplikasi



Gambar 5.10 Halaman awal aplikasi ketika menampilkan daftar kontak



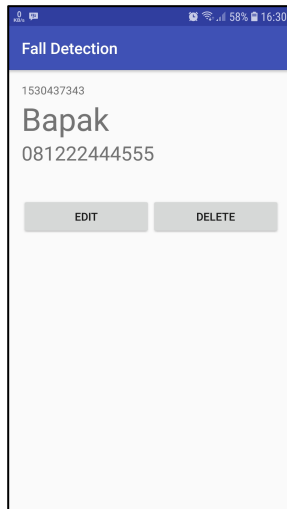
Gambar 5.11 Halaman tambah kontak



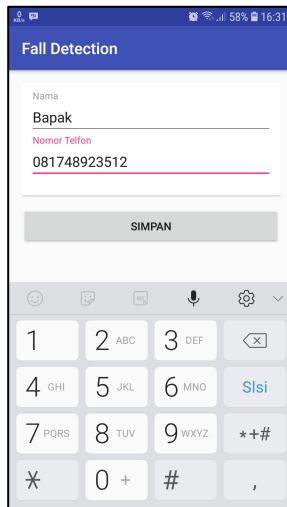
Gambar 5.12 Halaman awal ketika data kontak berhasil disimpan

2. Uji coba mengubah kontak

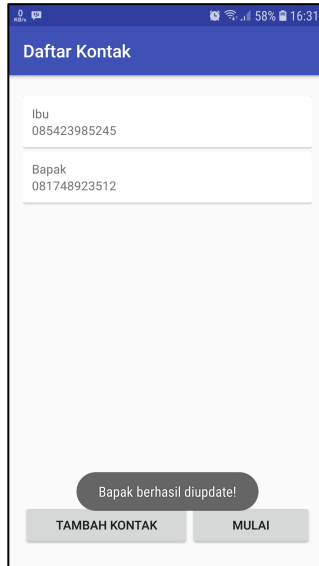
Fitur mengubah kontak pada aplikasi ini digunakan untuk melakukan perubahan data kontak berupa nama atau nomor telepon. Untuk mengubah kontak, pengguna perlu menekan salah satu kontak yang ada pada daftar kontak yang ada pada halaman awal aplikasi. Kemudian pengguna akan diarahkan ke halaman lihat kontak. Selanjutnya pengguna perlu menekan tombol edit dan pengguna akan diarahkan ke halaman untuk mengubah kontak. Halaman untuk mengubah kontak berisi konten yang sama dengan halaman tambah kontak, yaitu *form* isian untuk mengubah nama dan nomor telepon. Setelah itu pengguna menekan tombol simpan kemudian pengguna akan kembali diarahkan ke halaman awal aplikasi.



Gambar 5.13 Halaman lihat kontak



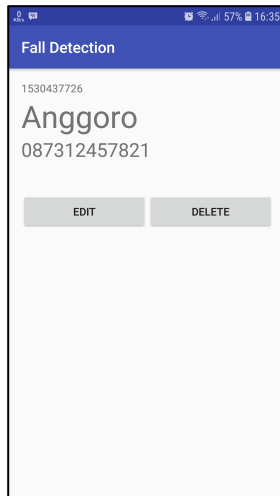
Gambar 5.14 Halaman untuk mengubah kontak



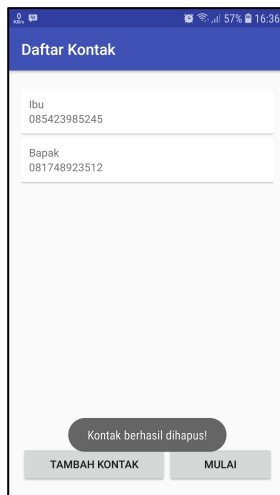
Gambar 5.15 Halaman awal ketika berhasil mengubah kontak

3. Uji coba menghapus kontak

Fitur hapus kontak berfungsi untuk menghapus kontak dari daftar kontak yang akan dikirimkan notifikasi berupa sms jika pengguna terdeteksi jatuh. Untuk menghapus kontak, pengguna diperlukan menekan salah satu kontak yang ingin dihapus pada daftar kontak. Selanjutnya pengguna akan diarahkan ke halaman lihat kontak seperti pada saat pengguna ingin mengubah kontak. Setelah itu, pengguna cukup menekan *delete* dan data sudah terhapus ketika pengguna sudah diarahkan kembali ke halaman awal aplikasi.



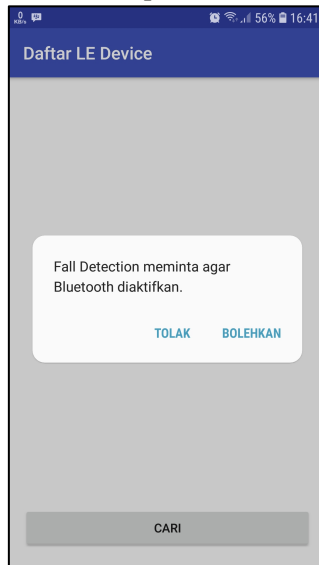
Gambar 5.16 Halaman lihat kontak



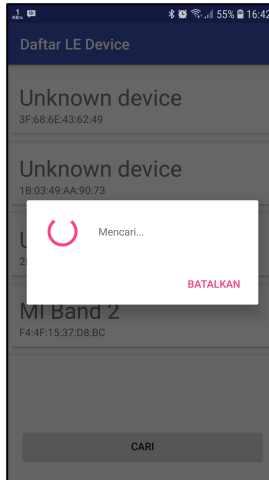
Gambar 5.17 Halaman awal ketika pengguna berhasil menghapus kontak

4. Uji coba mencari perangkat *bluetooth*

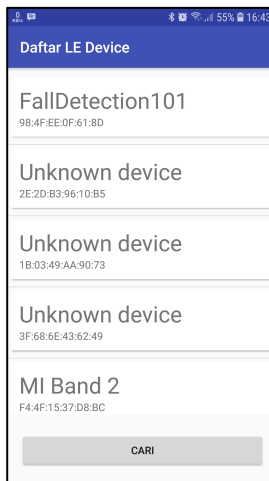
Pencarian perangkat *bluetooth* dilakukan ketika pengguna menekan tombol mulai pada halaman awal aplikasi. Kemudian pengguna akan diarahkan ke halaman pencarian perangkat *bluetooth*. Secara *default* aplikasi akan otomatis melakukan pencarian perangkat *bluetooth* ketika pengguna masuk ke halaman pencarian perangkat *bluetooth*. Akan tetapi pengguna bisa melakukan pencarian manual dengan menekan tombol cari. Setelah pencarian selesai, aplikasi akan menampilkan daftar perangkat *bluetooth* dari hasil pencarian.



Gambar 5.18 Halaman pencarian perangkat *bluetooth* ketika meminta aktivasi *bluetooth*



Gambar 5.19 Halaman pencarian perangkat *bluetooth* ketika melakukan pencarian perangkat

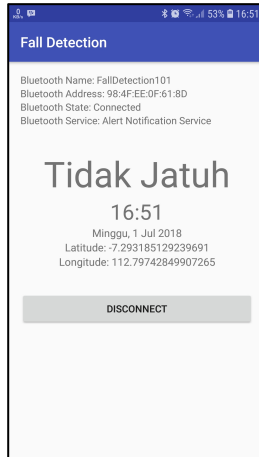


Gambar 5.20 Halaman pencarian perangkat *bluetooth* ketika menampilkan daftar perangkat hasil pencarian

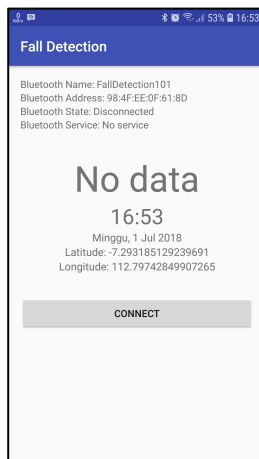
5. Uji coba proses deteksi jatuh

Untuk memulai proses *fall detection*, pengguna perlu menekan tombol mulai pada halaman awal aplikasi. Setelah itu, pengguna akan diarahkan ke halaman pencarian perangkat *bluetooth*. Jika perangkat *bluetooth* sudah terdeteksi, maka pengguna cukup menekan salah satu perangkat yang ditampilkan pada daftar perangkat *bluetooth* tersebut. Kemudian pengguna akan diarahkan ke halaman *fall detection*. Secara *default* perangkat android akan melakukan koneksi dengan perangkat *bluetooth* secara otomatis ketika pengguna memasuki halaman *fall detection*, tetapi jika gagal dalam proses koneksi, pengguna dapat menekan tombol *connect* untuk melakukan koneksi dengan perangkat *bluetooth*. Pengguna juga bisa memutuskan koneksi dengan perangkat *bluetooth* dan mengakhiri proses deteksi jatuh dengan cara menekan tombol *disconnect*. Jika perangkat android sudah terkoneksi maka tombol yang muncul adalah tombol *disconnect*, tetapi jika perangkat android belum terkoneksi dengan perangkat *bluetooth*, tombol yang muncul adalah tombol *connect* untuk melakukan koneksi perangkat android dengan perangkat *bluetooth* dan memulai proses deteksi jatuh.

Pada halaman *fall detection* berisi konten data perangkat *bluetooth* seperti nama perangkat *bluetooth*, alamat perangkat *bluetooth*, status koneksi perangkat *bluetooth*, nama *service* perangkat *bluetooth* dan yang paling penting adalah data dari *microcontroller arduino* yang dikirim melalui perangkat *bluetooth* tersebut. Selain itu terdapat juga data waktu, tanggal, dan data *latitude*, *longitude* dari lokasi perangkat android.



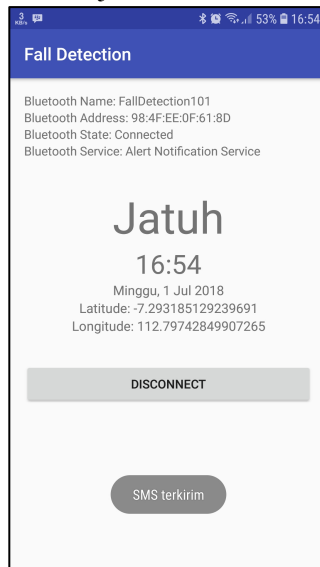
Gambar 5.21 Halaman *fall detection* ketika perangkat android sudah terkoneksi dengan perangkat *bluetooth* dan sedang melakukan proses deteksi jatuh



Gambar 5.22 Halaman *fall detection* ketika perangkat android tidak terkoneksi dengan perangkat *bluetooth*

6. Uji coba pengiriman notifikasi berupa sms

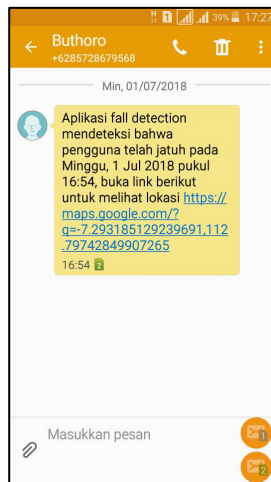
Pengiriman notifikasi email berupa sms dilakukan ketika pengguna terdeteksi jatuh. Aplikasi akan mengirimkan sms secara otomatis kepada keluarga / kerabat pengguna yang kontakannya sudah terdaftar dalam aplikasi sesaat setelah pengguna terdeteksi jatuh. Sms notifikasi ini berisi pemberitahuan bahwa pengguna telah terdeteksi jatuh beserta tanggal, waktu dan tautan lokasi ketika pengguna terdeteksi jatuh.



Gambar 5.23 Halaman *fall detection* ketika pengguna terdeteksi jatuh dan aplikasi mengirimkan notifikasi berupa sms



Gambar 5.24 Sms terkirim yang ditampilkan melalui perangkat android pengguna



Gambar 5.25 Sms diterima yang ditampilkan melalui perangkat android keluarga / kerabat pengguna



Gambar 5.26 Hasil tautan lokasi jatuh pengguna yang dibuka melalui perangkat android keluarga / kerabat pengguna

5.3.2 Uji Coba Performa Sistem

Uji coba performa digunakan untuk mengetahui seberapa besar tingkat akurasi sistem yang telah diimplementasikan. Uji coba performa ini dilakukan secara *offline*. Data yang didapatkan dari sensor *accelerometer* dan *gyroscope* ditampung dan diolah secara manual dengan menggunakan Microsoft Excel. Pengujian performa ini dilakukan secara *offline* dengan dasar bahwa performa sistem lebih diutamakan dalam penelitian ini. Uji coba performa sistem ini akan memperagakan gerakan jatuh dan memperagakan aktivitas sehari – hari yang memiliki kecenderungan yang mendekati sama dengan gerakan jatuh maupun yang tidak memiliki kemiripan dengan gerakan jatuh. Posisi pemasangan sensor pada pengguna dapat dilihat pada **Gambar 5.27**.



Gambar 5.27 Posisi pemasangan perangkat sensor pada pengguna

Uji coba performa sistem akan dibagi menjadi dua skenario dengan skenario pertama adalah memperagakan gerakan jatuh dan skenario yang kedua adalah memperagakan aktivitas sehari – hari yang memiliki kemiripan dengan jatuh maupun aktivitas yang tidak memiliki kemiripan dengan jatuh.

1. Skenario Uji Coba 1

Skenario uji coba satu dilakukan dengan memperagakan beberapa gerakan jatuh. Skenario ini dilakukan untuk menguji sensitivitas dari sistem *fall detection* menggunakan metode *quad threshold* ketika mendeteksi gerakan jatuh pada pengguna. Sensitivitas sistem digunakan untuk mengetahui seberapa besar sistem ketika mendeteksi positif (jatuh) pada saat pengguna memperagakan gerakan jatuh. Uji coba dilakukan oleh pengguna dengan memperagakan gerakan jatuh di atas kasur dengan jumlah 30 kali percobaan dengan arah jatuh yang bervariasi. Arah jatuh tersebut dibagi menjadi tiga arah jatuh, yaitu jatuh ke belakang,

jatuh ke samping kanan, dan jatuh ke samping kiri. Arah jatuh ke belakang dipilih dan diutamakan pada uji coba karena jatuh ke belakang dinilai sebagai jatuh yang memiliki risiko yang paling berbahaya. Sedangkan jatuh ke samping kanan dan kiri juga dilakukan uji coba untuk mengetahui seberapa besar sistem ini dapat mendeteksi jatuh selain jatuh ke belakang. Untuk menghitung sensitivitas pada sistem *fall detection* ini diperlukan nilai *true positive* (TP) dan *false negative* (FN). Nilai *true positive* didapat dari uji coba gerakan jatuh dan hasil uji coba mendeteksi bahwa pengguna jatuh. Sedangkan nilai *false negative* didapat dari uji coba gerakan jatuh dan hasil uji coba mendeteksi bahwa pengguna tidak jatuh. **Tabel 5.2** menunjukkan hasil deteksi jatuh pada skenario uji coba satu.

Tabel 5.2 Hasil uji coba jatuh pada skenario uji coba 1

No.	Aktivitas	Jumlah	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
1	Jatuh ke belakang	10	9	1
2	Jatuh ke kanan	10	9	1
3	Jatuh ke kiri	10	7	3
Total		30	25	5

Dari **Tabel 5.2** dapat diketahui bahwa pengguna memperagakan gerakan jatuh sebanyak 30 kali uji coba, 10 kali uji coba jatuh ke belakang, 10 kali uji coba jatuh ke kanan dan 10 kali uji coba jatuh ke kiri. Ketika pengguna memperagakan jatuh ke belakang dari 10 uji coba, sistem mendeteksi jatuh 9 kali dan gagal mendeteksi jatuh 1 kali. Hasil yang sama juga diperoleh pada uji

coba pada saat pengguna memperagakan jatuh ke kanan. Sedangkan pada saat pengguna memperagakan gerakan jatuh ke kiri dari 10 kali percobaan, sistem mendeteksi jatuh 7 kali dan gagal mendeteksi jatuh 3 kali. Dengan hasil total yang diperoleh pada **Tabel 5.2** dapat dihitung nilai sensitivitas pada sistem *fall detection* ini.

$$\begin{aligned} \text{Sensitivitas} &= \frac{TP}{TP + FN} \times 100\% \\ \text{Sensitivitas} &= \frac{25}{27 + 3} \times 100\% \\ \text{Sensitivitas} &= 83,33\% \end{aligned}$$

2. Skenario Uji Coba 2

Skenario uji coba dua dilakukan dengan memperagakan beberapa aktivitas sehari – hari seperti jalan, duduk dan merebahkan badan. Skenario ini dilakukan untuk menguji spesifitas dari sistem *fall detection* menggunakan metode *quad threshold* ketika pengguna sedang melakukan aktivitas sehari – hari. Spesifitas sistem digunakan untuk mengetahui seberapa baik sistem mengidentifikasi negatif (tidak jatuh) pada saat pengguna memperagakan gerakan aktivitas sehari – hari. Uji coba dilakukan oleh pengguna dengan memperagakan aktivitas jalan sebanyak 5 kali, aktivitas duduk sebanyak 5 kali dan aktivitas merebahkan badan sebanyak 20 kali. Uji coba gerakan merebahkan badan dilakukan karena gerakan tersebut memiliki nilai *accelerometer* dan *gyroscope* yang hampir sama dengan gerakan jatuh. Sedangkan uji coba duduk dan berjalan dilakukan untuk membuktikan bahwa gerakan duduk dan berjalan memiliki nilai *accelerometer* dan *gyroscope* yang berbeda jauh dibandingkan dengan jatuh. Untuk menghitung spesifitas pada sistem *fall detection* ini diperlukan nilai *false positive* (FP) dan *true negative* (TN). Nilai *false psotive* didapat dari hasil uji coba pada saat pengguna melakukan aktivitas sehari – hari dan hasil uji coba mendeteksi bahwa pengguna jatuh. Sedangkan *true negative* didapat dari hasil uji coba pada saat pengguna melakukan aktivitas

sehari – hari dan hasil uji coba mendeteksi bahwa pengguna tidak jatuh. **Tabel 5.3** menunjukkan hasil skenario uji coba dua pada saat pengguna mempragakan beberapa aktivitas sehari – hari.

Tabel 5.3 Hasil uji coba mempragakan aktivitas sehari - hari pada skenario uji coba 2

No.	Aktivitas	Jumlah	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>
1	Merebahkan badan	20	2	18
2	Duduk	5	0	5
3	Jalan	5	0	5
Total		30	2	28

Pengguna mempragakan aktivitas sehari – hari sebanyak 30 kali percobaan yang masing – masing adalah 20 kali uji coba merebahkan badan, 5 kali uji coba duduk, dan 5 kali uji coba jalan. Ketika pengguna mempragakan 20 kali percobaan merebahkan badan, 2 percobaan terdeteksi bahwa pengguna jatuh dan 18 percobaan tidak terdeteksi bahwa pengguna jatuh. Untuk aktivitas seperti duduk dan merebahkan badan, dari masing – masing 5 percobaan tidak ada yang terdeteksi jatuh. Dari nilai total pada **Tabel 5.3** dapat dihitung nilai spesifitas pada sistem *fall detection* ini.

$$Spesifitas = \frac{TN}{FP + TN} \times 100\%$$

$$Spesifitas = \frac{28}{2 + 28} \times 100\%$$

$$Spesifitas = 93,33\%$$

5.3.3 Analisis Hasil Uji Coba

Berdasarkan hasil uji coba pada skenario uji coba satu dan skenario uji coba dua didapatkan empat nilai yang dapat digunakan untuk menghitung performa sistem ketika mendeteksi jatuh pada pengguna. Berikut adalah nilai – nilai yang didapatkan dari skenario uji coba satu dan skenario uji coba dua:

1. *True positive* (TP) = 25
2. *False negative* (FN) = 5
3. *False positive* (FP) = 2
4. *True negative* (TN) = 28

Dari nilai – nilai tersebut dapat digunakan untuk menghitung nilai sensitivitas, nilai spesifitas, nilai prediksi positif (NPP), nilai prediksi negatif (NPN), dan akurasi pada sistem *fall detection* menggunakan metode *quad threshold*. Nilai prediksi positif dihitung dengan menggunakan rumus:

$$\text{Nilai prediksi positif} = \frac{TP}{TP + FP} \times 100\%$$

$$\text{Nilai prediksi positif} = \frac{25}{25 + 2} \times 100\%$$

$$\text{Nilai prediksi positif} = 92,59\%$$

Nilai prediksi positif memberitahukan kemungkinan seseorang jatuh jika hasil uji coba sistem mendeteksi jatuh. Sedangkan untuk menghitung nilai prediksi negatif digunakan rumus:

$$\text{Nilai prediksi negatif} = \frac{TN}{TN + FN} \times 100\%$$

$$\text{Nilai prediksi negatif} = \frac{28}{28 + 5} \times 100\%$$

$$\text{Nilai prediksi negatif} = 84,84\%$$

Nilai prediksi negatif memberitahukan kemungkinan seseorang tidak jatuh ketika hasil uji coba sistem mendeteksi tidak

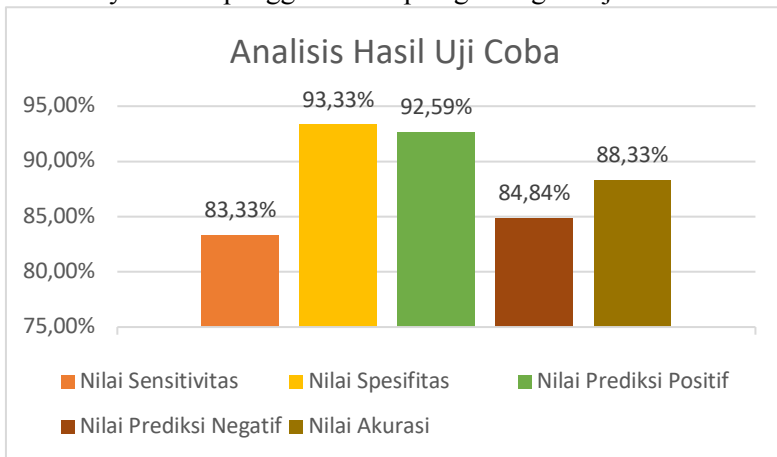
jatuh. Untuk mendapatkan nilai akurasi pada sistem *fall detection* dihitung dengan menggunakan rumus:

$$\text{Nilai akurasi sistem} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$\text{Nilai akurasi sistem} = \frac{25 + 28}{25 + 28 + 2 + 5} \times 100\%$$

$$\text{Nilai akurasi sistem} = 88,33\%$$

Nilai akurasi sistem memberitahukan seberapa besar tingkat kedekatan sistem mendeteksi jatuh dibandingkan dengan kondisi sebenarnya ketika pengguna memperagakan gerak jatuh.



Gambar 5.28 Analisis hasil uji coba

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Pada penelitian ini dibangun sistem *monitoring* dan notifikasi *fall detection* dengan metode *quad threshold* berbasis *dual sensor* akselerasi dan orientasi *multiposisi*. Sistem ini menggunakan nilai yang didapatkan dari masing – masing dua sensor *accelerometer* dan *gyroscope* untuk menentukan gerakan jatuh pengguna.
2. Sistem *fall detection* dengan menggunakan metode *quad threshold* memiliki nilai sensitivitas 83,33%, nilai spesifitas 93,33%, nilai prediksi positif 92,59%, nilai prediksi negatif 84,84%, dan nilai akurasi sebesar 88,33%.
3. Ketika pengguna terdeteksi jatuh, sistem akan mengirimkan notifikasi berupa sms untuk keluarga dan kerabat pengguna. Notifikasi tersebut berisi pemberitahuan bahwa pengguna sedang jatuh dan disertai lokasi dimana pengguna tersebut berada.

6.2 Saran

Saran yang dapat diberikan dalam pengujian *fall detection* dengan metode *quad threshold* adalah sebagai berikut:

1. Menambah jumlah skenario pengujian dan menambah gerakan yang diperagakan, karena sistem ini butuh berbagai macam kondisi pengguna untuk benar – benar bisa menentukan jatuh pengguna dengan tingkat sensitivitas, spesifitas dan akurasi yang tinggi.
2. Perlu adanya media untuk sarana mengirim notifikasi yang lebih *portable* karena dirasa jika menggunakan *smartphone* masih terlalu banyak barang yang harus dibawa oleh pengguna.

DAFTAR PUSTAKA

- [1] “Masalah Khusus Dan Masalah Kesehatan Yang Terjadi Lanjut Usia,” 16 Oktober 2012. [Online]. Available: <https://www.scribd.com/document/110163876/Masalah-Khusus-Dan-Masalah-Kesehatan-Yang-Terjadi-Lanjut-Usia>. [Diakses 25 Desember 2017].
- [2] M. Kangas, A. Konttila dan I. Winblad, “Determination of Simple Thresholds for Accelerometry-Based Parameters for Fall Detection,” dalam *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Lyon, France, 2007.
- [3] M. J. Mathie, J. Basilakis dan B. G. Celler, “A System for Monitoring Posture and Physical Activity Using Accelerometer,” dalam *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Istanbul, Turkey, 2001.
- [4] D. Brulin dan E. Courtial, “Multi-Sensor Data Fusion System for Fall Detection,” dalam *10th IEEE International Conference on Information Technology and Applications in Biomedicine*, Corfu, Greece, 2010.
- [5] J. Jacob, T. Nguyen, D. Y. Lie, S. Zupancic, J. Bishara, A. Dentino dan R. E. Banister, “A Fall Detection Study on the Sensors Placement Location and a Rule-Based Multi-Thresholds Algorithm Using Both Accelerometer and Gyroscopes,” dalam *IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, 2011.
- [6] A. K. Bourke dan G. M. Lyons, “A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor,” dalam *Medical Engineering & Physics*, 2008.
- [7] “Arduino - Introduction,” [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Diakses 19 Desember 2017].

- [8] "What is an Arduino? - learn.sparkfun.com," [Online]. Available: <https://learn.sparkfun.com/tutorials/what-is-an-arduino>. [Diakses 19 Desember 2017].
- [9] "Arduino Comparison Guide," [Online]. Available: <https://learn.sparkfun.com/tutorials/arduino-comparison-guide>. [Diakses 05 Juni 2018].
- [10] "Arduino - Environment," 07 September 2015. [Online]. Available: <https://www.arduino.cc/en/Guide/Environment>. [Diakses 19 Desember 2017].
- [11] "Apa itu Accelerometer Sensor? - Mantapps.co.id," 05 Desember 2015. [Online]. Available: <http://www.mantapps.co.id/news/tips-tricks/apa-itu-accelerometer-sensor/>. [Diakses 19 Desember 2017].
- [12] "Apa itu Gyroscope? - Mantapps.co.id," 05 Desember 2015. [Online]. Available: <http://www.mantapps.co.id/news/tips-tricks/apa-itu-gyroscope/>. [Diakses 19 Desember 2017].
- [13] "MPU-6050 Accelerometer + Gyro," [Online]. Available: <https://playground.arduino.cc/Main/MPU-6050>. [Diakses 05 Juni 2018].
- [14] "MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Devices," [Online]. Available: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>. [Diakses 05 Juni 2018].
- [15] "ELEKTRO INDONESIA - BLUETOOTH: Teknologi Komunikasi Wireless untuk Layanan Multimedia dengan Jangkauan Terbatas," April 2001. [Online]. Available: <http://www.elektroindonesia.com/elektro/khu36.html>. [Diakses 19 Desember 2017].
- [16] "Sejarah Lahirnya Android," 07 Oktober 2014. [Online]. Available: https://www.kompasiana.com/fariseiffel/sejarah-lahirnya-android_54f963e8a33311b6078b4e82. [Diakses 01 Januari 2018].

- [17] “Arti Android Beserta Fasilitas Didalamnya - Info Teknologi,” [Online]. Available: <http://www.infoteknologi.com/apa-itu-android/>. [Diakses 01 Januari 2018].
- [18] “Meet Android Studio | Android Studio,” [Online]. Available: <https://developer.android.com/studio/intro/index.html>. [Diakses 19 Desember 2017].
- [19] “Pengenalan, Pengertian dan Memahami Android Studio Lebih Dekat - Komputerdia | Berbagi Tutorial,” 24 Agustus 2017. [Online]. Available: <http://www.komputerdia.com/2017/08/pengenalan-dan-memahami-android-studio-lebih-dekat.html>. [Diakses 19 Desember 2017].

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Tabel 6.1 Data *training* aktivitas berjalan pada sensor 2

ax	ay	az	gx	gy	gz
0,17	0,96	0,08	2,13	-17,57	1,46
0,04	0,89	0,14	-11,59	-2,98	-10,28
0,21	0,90	0,10	2,72	10,53	0,61
0,08	0,93	0,15	-0,60	-11,27	2,55
0,09	0,93	0,10	0,52	0,76	0,05
0,05	0,92	0,17	-3,17	-0,29	-3,87
-0,12	0,95	0,10	-6,63	-38,06	-34,70
0,20	0,97	0,19	-8,60	-28,66	-12,63
-0,04	0,92	0,25	-24,05	-17,47	-3,94
-0,06	0,85	0,24	-12,27	-32,00	-5,66
0,03	1,11	0,25	15,79	-62,93	-9,27
0,08	0,92	0,15	39,95	-0,28	1,48
-0,03	0,93	0,11	32,11	15,01	-1,63
-0,09	0,94	-0,04	34,09	46,63	7,65
0,01	0,93	-0,11	25,28	31,62	11,01
0,01	1,04	0,05	12,16	55,86	15,87
0,02	1,05	0,17	-26,29	17,01	-5,39
-0,11	1,08	-0,20	-55,36	-45,51	-28,52
0,08	0,89	0,03	-62,22	-48,33	-9,82
-0,05	0,85	0,11	-0,22	-103,47	-17,33
0,18	0,82	0,47	-5,95	-117,48	-11,89
-0,04	0,85	-0,02	49,73	-0,30	-17,24
-0,30	0,92	-0,11	62,51	81,09	-7,14
-0,14	0,94	-0,12	42,91	26,01	12,95

ax	ay	az	gx	gy	gz
-0,08	1,00	0,05	24,26	61,60	9,98
0,02	1,02	0,40	-67,66	19,03	-22,60
0,19	0,97	0,02	-68,37	-167,31	-28,44
0,06	0,79	0,05	-41,90	-57,14	2,66
-0,09	0,91	0,23	21,76	3,09	-14,79
0,24	0,88	0,26	-1,88	3,81	-2,04
0,32	0,89	0,01	53,30	23,66	-3,50
0,05	0,94	0,13	33,65	48,72	-6,47
-0,08	0,91	-0,06	42,34	73,05	5,95
-0,08	0,95	-0,07	21,15	88,02	9,73
-0,04	1,04	-0,03	-3,85	107,27	3,73
0,12	1,09	-0,18	-124,65	121,73	-17,20
0,14	0,86	-0,02	-44,82	113,16	4,34
-0,03	0,78	0,43	-18,78	43,84	15,32
-0,13	0,81	0,33	10,76	35,34	8,53
0,05	0,96	0,36	34,21	74,88	17,84
0,01	0,85	0,00	35,81	74,93	-2,86
-0,13	0,94	-0,05	28,79	106,41	-3,89
-0,09	0,97	-0,03	17,47	72,50	4,21
-0,13	0,95	-0,10	16,44	85,87	12,76
-0,13	0,95	-0,02	15,25	66,30	16,82
0,17	1,05	0,27	-19,00	53,84	-0,22
-0,21	1,01	-0,13	-68,71	12,91	-39,37
0,07	0,87	0,01	-50,81	1,08	-6,65
-0,18	0,83	0,38	-29,30	-3,92	-7,71
-0,13	0,78	0,42	-1,53	-2,54	0,23

ax	ay	az	gx	gy	gz
0,05	0,86	0,40	16,70	-29,85	8,79
0,07	1,06	0,55	-34,98	-18,57	-16,87
0,15	0,93	0,09	50,38	22,21	2,63
-0,03	0,93	0,11	30,92	14,76	3,40
-0,07	0,93	-0,05	41,52	64,55	16,34
-0,14	1,16	0,10	13,77	61,69	11,43
0,33	1,10	0,03	-116,08	37,13	-18,19
0,01	0,83	0,10	-37,98	-11,98	-13,57
0,13	0,82	0,29	-20,31	-32,76	-0,86
-0,02	0,91	0,42	-23,60	-14,36	1,34
-0,11	0,78	0,33	7,97	-18,24	-7,22
0,02	0,53	-0,16	52,69	116,44	-19,97
0,02	0,93	0,16	36,11	56,16	1,91
-0,09	0,92	-0,01	42,80	28,15	7,78
-0,05	0,95	0,00	16,93	51,48	5,85
-0,08	0,91	-0,10	25,10	37,52	15,59
-0,15	0,98	-0,04	14,82	40,02	17,75
0,04	1,02	0,16	-45,95	48,50	-13,38
0,17	0,99	0,04	-50,13	-91,70	-26,50
0,07	0,84	0,04	-27,14	-46,79	-13,47
-0,03	0,84	0,31	-13,77	-39,51	-17,18
0,11	0,89	0,27	-8,73	-12,42	-3,73
-0,04	1,10	0,43	11,44	-81,53	-3,29
-0,16	0,93	0,17	37,22	-35,51	-0,61
-0,04	0,94	0,06	27,60	47,94	-4,88
-0,02	0,92	-0,03	45,27	27,12	2,05

ax	ay	az	gx	gy	gz
-0,08	0,97	-0,06	15,15	61,20	1,67
-0,03	0,92	-0,08	13,05	73,56	6,27
-0,14	0,95	-0,09	-26,53	140,88	-9,64
0,19	1,05	-0,05	-83,64	120,24	-7,85
0,01	0,85	0,07	-41,72	99,69	20,06
0,02	0,80	0,23	-6,17	78,93	23,70
-0,29	0,88	0,45	16,69	19,25	-5,46
-0,01	0,86	0,06	13,80	50,66	-1,65
0,05	0,92	0,05	18,46	43,65	2,40
-0,11	0,96	-0,01	17,14	47,71	-3,21
-0,17	0,95	0,00	6,73	40,70	3,85
-0,17	0,92	0,05	7,82	67,26	9,33
-0,20	0,94	0,02	13,41	76,66	19,00
0,10	1,09	-0,02	-15,57	82,05	12,92
0,22	1,02	-0,03	-68,97	40,79	-14,79
-0,03	0,84	0,06	-5,37	11,60	3,30
0,17	0,97	0,42	18,61	-26,30	-4,11
-0,15	0,89	0,11	14,66	12,18	8,04
-0,02	0,94	0,07	12,63	25,76	9,00
0,08	0,95	0,03	7,47	12,96	3,76
0,04	0,93	0,00	6,42	4,54	1,83
0,07	0,94	-0,02	3,86	5,02	-0,85
0,03	0,94	0,00	1,75	1,16	-0,42
0,04	0,94	0,00	2,59	1,13	-0,05
0,03	0,95	0,00	1,28	-0,59	-0,49
0,04	0,94	0,00	1,40	-1,92	-0,16

ax	ay	az	gx	gy	gz
0,03	0,94	0,00	1,66	-0,63	1,15
0,03	0,94	-0,02	-1,22	8,50	0,67
0,04	0,95	0,01	-0,67	-0,28	0,39
0,06	0,93	-0,03	-2,76	0,43	-0,81
0,05	0,94	0,01	-1,95	3,35	-0,35
0,04	0,94	0,01	-1,28	5,40	-0,19
0,03	0,94	0,03	0,29	-4,57	0,37
0,03	0,94	0,01	-1,15	2,26	-0,09
0,03	0,94	0,01	-1,63	-0,27	-0,66
0,06	0,94	-0,01	-0,39	3,46	-0,42

Tabel 6.2 Data *training* gerakan transisi pada sensor 2

ax	ay	az	gx	gy	gz
0,09	0,94	-0,05	-0,47	0,60	-0,87
0,07	0,94	-0,03	0,25	-2,77	-0,66
0,08	0,94	-0,05	-0,46	-9,07	-1,75
0,08	0,94	-0,04	-0,49	-0,39	0,50
0,09	0,94	-0,04	-0,85	0,04	-0,45
0,10	0,94	-0,03	-0,19	-0,44	0,34
0,09	0,94	-0,04	0,12	0,51	0,35
0,10	0,94	-0,04	0,08	0,79	0,54
0,09	0,94	-0,05	-2,20	1,99	-1,06
0,07	0,93	-0,04	-0,56	1,10	-0,70
0,08	0,94	-0,03	-1,78	-0,76	-2,30
0,07	0,94	-0,02	-7,80	-0,60	-6,89
0,03	0,94	0,03	-12,33	-21,05	-6,34

ax	ay	az	gx	gy	gz
-0,04	0,94	0,11	-23,07	-29,07	-11,67
0,05	0,92	0,20	-64,31	-33,58	1,78
-0,09	0,80	0,29	-74,66	51,75	-4,11
-0,16	0,87	0,47	-58,97	-9,89	-23,59
0,04	0,74	0,62	-52,95	-5,18	-10,84
-0,07	0,60	0,78	-53,66	6,37	1,95
-0,19	0,41	0,89	-52,12	21,34	-2,85
-0,16	0,50	0,81	-34,06	10,84	3,85
-0,16	0,34	0,91	-42,24	7,98	-0,84
-0,16	0,16	0,96	-54,51	8,63	1,73
-0,18	0,05	0,95	-59,71	1,57	-8,18
-0,28	-0,03	0,88	-75,08	13,37	-5,71
-0,34	-0,31	0,99	-93,35	1,58	-27,47
-0,03	-0,52	1,14	-40,74	-28,37	0,77
-0,10	-0,65	0,79	-39,71	-3,18	13,78
-0,34	-0,83	0,61	-31,98	8,08	22,89
-0,38	-0,75	0,67	29,40	-0,76	-4,99
-0,19	-0,54	0,72	51,21	-24,22	-3,49
-0,27	-0,37	0,86	80,28	-6,46	7,08
0,37	-0,07	1,23	24,89	-103,31	-9,02
-0,32	-0,22	0,83	25,06	7,11	-2,72
0,05	-0,19	1,10	20,16	18,45	4,81
-0,06	-0,09	0,95	8,05	17,76	17,92
-0,16	-0,04	1,11	0,99	4,69	-1,18
-0,07	-0,10	0,97	-1,57	-9,30	-1,76
-0,10	-0,09	1,02	-2,36	-3,31	1,83

ax	ay	az	gx	gy	gz
-0,09	-0,09	1,01	-3,05	-2,60	-0,76
-0,06	-0,11	0,99	2,17	-1,53	-1,34
-0,09	-0,09	1,03	-2,31	-0,73	1,12
-0,09	-0,11	1,01	-0,67	-0,17	0,23
-0,08	-0,10	0,99	-1,21	-1,90	-0,98
-0,10	-0,11	1,02	-1,18	0,50	0,37
-0,09	-0,12	0,98	-1,21	-1,21	0,27
-0,08	-0,10	1,02	-0,27	-2,79	0,15
-0,08	-0,12	1,01	1,17	-2,37	0,52
-0,07	-0,11	1,02	0,80	-1,97	0,25
-0,07	-0,11	0,99	-0,47	0,84	-0,14
-0,09	-0,11	1,01	0,60	-1,52	0,73
-0,06	-0,11	1,00	-0,11	-0,50	0,08
-0,07	-0,10	1,00	0,57	0,31	-0,80
-0,06	-0,10	1,00	-0,87	0,81	0,70
-0,10	-0,08	1,03	-3,85	0,24	-0,78
-0,06	-0,18	0,98	0,04	-10,24	-1,14
0,00	-0,11	0,98	5,22	-2,55	1,53
-0,03	-0,11	1,00	2,26	0,95	1,77
0,06	-0,06	1,10	-1,01	10,11	6,27
-0,10	-0,12	1,05	-28,11	9,15	6,31
-0,19	-0,17	1,06	-44,31	10,41	2,20
-0,19	-0,34	0,96	-54,16	41,01	5,18
-0,20	-0,41	0,93	-46,02	28,05	18,49
-0,37	-0,58	0,81	-25,00	35,63	6,13
-0,48	-0,65	0,83	-23,95	-17,78	9,87

ax	ay	az	gx	gy	gz
-0,43	-0,58	0,73	3,82	20,79	-23,47
-0,34	-0,64	0,77	13,06	-30,77	-10,88
-0,18	-0,47	0,75	22,50	5,06	9,56
-0,24	-0,43	0,95	4,26	-18,15	5,69
-0,25	-0,48	0,88	-0,45	-21,48	4,66
-0,30	-0,57	0,90	1,46	-17,12	-9,92
-0,32	-0,58	0,90	21,02	37,88	-4,98
-0,23	-0,34	1,06	79,75	-25,08	32,54
-0,12	-0,12	1,07	85,26	-16,08	24,61
-0,17	0,11	0,99	98,11	-18,46	10,50
-0,09	0,42	0,86	108,75	-38,23	-2,29
0,00	0,54	0,61	107,73	-40,55	3,10
0,05	0,62	0,56	76,35	-21,85	6,42
0,12	0,73	0,42	55,81	20,79	3,08
0,27	0,92	0,32	42,51	78,21	6,69
0,00	0,85	0,26	-8,64	-42,61	-3,15
0,24	0,75	0,12	44,95	-18,82	1,95
0,34	0,90	-0,05	32,85	71,90	26,57
0,11	0,93	-0,01	17,27	-17,56	-10,93
0,05	0,92	-0,06	8,94	-10,22	-7,71
0,07	0,98	0,01	3,89	-23,41	-6,55
-0,09	1,01	-0,09	-1,95	-1,48	0,34
-0,07	0,98	-0,01	2,95	-32,13	1,28
-0,01	0,95	-0,08	3,50	5,47	5,92
-0,05	0,96	0,03	1,24	-3,09	5,75
0,12	0,94	-0,08	-2,24	23,53	6,07

ax	ay	az	gx	gy	gz
0,12	0,93	0,00	-2,00	5,16	2,89
0,13	0,93	-0,02	-0,19	1,09	1,24
0,12	0,94	0,00	-2,56	2,85	0,05
0,14	0,94	0,02	0,53	0,43	0,62
0,12	0,94	0,00	-0,97	-2,82	0,47
0,13	0,93	-0,01	1,17	-0,69	-0,30
0,13	0,93	0,00	2,31	1,46	0,18
0,13	0,93	-0,01	0,92	0,03	-0,41
0,14	0,93	-0,02	0,44	0,44	-0,96
0,11	0,94	-0,01	1,19	-0,31	-0,51
0,11	0,94	-0,01	1,06	-2,30	0,69
0,12	0,94	-0,02	-0,08	-2,81	0,22
0,12	0,93	-0,04	-0,22	2,76	-0,16
0,11	0,94	-0,03	0,94	2,32	-0,02
0,09	0,93	-0,02	0,63	-1,58	-0,03

Tabel 6.3 Data *training* gerakan merebahkan badan pada sensor 1

ax	ay	az	gx	gy	gz
-0,18	0,91	-0,15	-1,53	2,52	0,15
-0,20	0,92	-0,19	-1,03	-1,77	0,82
-0,14	0,87	-0,20	-8,92	-7,76	0,81
-0,20	0,93	-0,16	-21,02	-7,87	-5,89
-0,21	0,91	-0,12	-20,13	-6,25	-4,02
-0,23	0,89	-0,03	-31,32	-5,61	-13,93
-0,26	0,85	0,07	-50,54	-20,13	-17,30

-0,27	0,72	0,34	-94,20	-21,73	-20,47
-0,27	0,50	0,50	-141,87	-31,59	-26,34
-0,45	0,28	0,97	-235,61	-60,82	-35,75
0,07	-0,50	0,45	-29,11	-29,96	-12,07
-0,25	0,12	0,96	5,86	10,77	6,47
-0,19	0,01	0,98	-7,70	1,89	1,42
-0,21	-0,04	0,99	0,84	-6,61	-0,86
-0,22	0,01	1,01	-1,37	-0,50	-2,33
-0,16	0,00	0,95	12,40	3,37	0,64
-0,19	-0,03	0,94	7,18	13,37	5,08
-0,19	0,04	0,99	-3,32	4,08	10,03
-0,29	-0,12	1,02	6,14	8,10	8,09
-0,32	0,08	1,03	54,94	8,30	6,33
-0,28	0,23	0,93	68,61	8,64	30,40
-0,38	0,36	0,76	51,61	4,79	17,32
-0,36	0,55	0,72	47,94	7,63	22,73
-0,19	0,77	0,61	65,26	5,60	20,40
-0,15	0,81	0,29	57,23	-10,36	10,96
-0,16	0,90	0,13	39,92	-2,54	2,41
-0,19	0,86	0,09	15,95	11,18	7,69
-0,13	0,93	0,05	8,05	10,57	-1,05
-0,16	0,95	0,10	4,20	-1,16	-2,38
-0,19	0,88	0,12	-2,85	1,27	0,37
-0,16	0,93	0,08	5,05	-3,88	-0,04
-0,13	0,93	0,03	4,03	-3,55	-3,63
-0,14	0,93	0,05	1,28	-2,96	-3,02
-0,19	0,92	0,08	-1,66	-3,35	-1,71

-0,18	0,93	0,06	-1,00	-2,03	-4,47
-------	------	------	-------	-------	-------

Tabel 6.4 Data *training* gerakan jatuh pada sensor 1

ax	ay	az	gx	gy	gz
-0,25	0,92	0,00	-0,73	-3,64	1,30
-0,20	0,97	0,01	-6,00	-2,50	4,06
-0,23	0,91	-0,01	-3,56	2,71	0,13
-0,27	0,94	0,01	-5,27	8,73	-0,37
-0,20	0,91	0,01	-4,37	1,82	3,76
-0,24	0,90	-0,06	2,73	2,44	0,85
-0,25	0,92	-0,04	2,92	0,36	0,08
-0,19	0,94	-0,06	-2,75	-4,55	5,53
-0,20	0,92	-0,06	6,86	3,92	8,40
-0,25	0,90	-0,16	14,97	7,82	13,10
-0,31	0,87	-0,14	-7,61	9,19	-0,44
-0,15	0,93	-0,25	-48,36	33,80	0,42
-0,14	0,88	-0,13	-5,89	4,87	-6,05
-0,19	0,93	-0,15	3,05	0,84	-2,98
-0,17	0,93	-0,14	-0,31	2,95	0,19
-0,21	0,91	-0,12	2,63	0,56	0,11
-0,19	0,92	-0,16	2,40	2,72	1,37
-0,19	0,92	-0,20	4,08	0,36	1,00
-0,20	0,91	-0,15	2,47	-5,17	1,59
-0,16	0,90	-0,18	-3,82	-5,04	-3,02
-0,18	0,85	-0,17	-1,98	7,38	1,48
-0,17	0,74	-0,14	23,61	-1,46	2,10
-0,03	0,68	-0,26	13,65	16,14	14,24

ax	ay	az	gx	gy	gz
-0,18	0,84	-0,52	9,72	-34,12	10,77
-0,10	0,85	-0,45	-39,97	-48,24	-9,63
-0,28	0,88	-0,39	-111,90	-15,20	-45,64
-0,54	1,71	0,59	25,42	-65,98	-47,11
0,36	-1,04	1,87	-250,14	-46,74	-22,22
0,00	0,00	0,00	0,00	0,00	0,00
-0,01	-0,15	1,06	6,27	11,22	-1,96
-0,05	0,16	1,05	1,73	17,05	-0,79
-0,13	0,09	1,03	25,90	3,69	5,28
-0,30	0,18	1,00	4,34	-5,03	2,48
-0,27	0,23	1,14	27,43	-11,59	6,99
-0,23	0,52	0,82	79,78	45,33	22,94
-0,27	0,63	0,58	65,73	6,50	23,92
-0,30	0,63	0,45	63,44	-18,92	21,37
-0,05	0,88	0,42	32,29	-7,20	10,85
-0,10	0,95	0,37	22,15	10,75	6,37
-0,13	0,88	0,35	22,61	16,15	-2,42
-0,14	0,99	0,33	28,92	2,73	-4,03
-0,13	0,91	0,13	21,85	-5,83	-3,54
-0,13	0,96	0,10	9,82	-8,90	-4,43
-0,17	0,89	0,04	18,11	6,24	-3,18
-0,20	0,88	0,14	0,81	1,80	0,17
-0,16	0,93	0,09	7,63	0,17	0,42
-0,17	0,90	0,17	-3,19	5,67	-1,35
-0,16	0,90	0,12	0,56	2,34	0,80
-0,19	0,92	0,12	-1,72	0,44	-1,45

ax	ay	az	gx	gy	gz
-0,14	0,97	0,04	0,79	11,01	-3,67
-0,13	0,99	0,07	-9,31	-1,31	-2,85
-0,21	0,93	0,10	-3,82	6,89	-2,55
-0,20	0,90	0,08	3,75	-1,92	-0,95
-0,20	0,89	0,07	2,69	1,95	-0,18
-0,24	0,90	0,13	-8,93	-3,60	0,91
-0,23	0,91	0,09	-5,03	-0,37	0,90
-0,21	0,92	0,10	0,25	-0,90	1,89
-0,20	0,93	0,12	-13,58	-9,28	3,53
-0,21	0,91	0,09	0,15	-0,69	2,86
-0,19	0,89	0,09	0,99	1,92	0,51
-0,19	0,94	0,06	-3,28	-1,56	-0,56
-0,18	0,93	0,10	-6,08	1,66	-0,47
-0,22	0,93	0,09	-2,15	5,60	-0,79
-0,25	0,93	0,14	-12,01	6,56	-1,60
-0,21	0,91	0,08	1,89	-0,28	-0,05

**Tabel 6.5 Contoh data uji coba aktivitas berjalan pada sensor
1**

ax	ay	az	gx	gy	gz
0,02	0,96	-0,03	0,95	-0,89	-1,58
0,02	0,95	-0,03	1,56	1,18	0,01
0,02	0,96	-0,02	-0,48	2,18	0,34
0,01	0,95	-0,05	5,44	3,44	1,77
-0,01	0,96	-0,02	2,92	2,56	2,66
0,04	0,96	-0,05	2,98	4,74	3,63

ax	ay	az	gx	gy	gz
0,04	0,96	-0,02	0,46	1,20	4,04
0,07	0,95	-0,06	6,65	4,68	6,76
0,10	0,99	-0,01	2,45	15,12	4,91
0,14	0,93	-0,01	6,89	5,20	0,11
0,12	0,95	0,04	-13,39	-15,63	-15,03
0,13	1,07	-0,08	-1,79	5,59	-5,69
-0,04	0,92	-0,14	1,64	3,73	-0,94
0,01	0,92	-0,07	10,57	10,46	3,00
0,03	0,97	-0,04	5,84	32,47	-1,13
0,05	1,13	-0,11	-10,05	50,05	6,96
0,17	0,83	-0,18	-1,24	45,73	3,07
0,31	0,90	0,07	1,68	32,67	4,62
0,09	0,89	0,04	1,03	-9,08	-6,72
0,07	1,19	-0,05	-17,08	-31,03	-11,19
-0,01	0,95	-0,24	15,87	-51,03	6,46
-0,06	0,87	-0,17	18,50	-33,21	5,09
0,08	0,98	-0,14	7,05	1,17	6,13
0,10	1,00	-0,04	-3,90	-101,92	-11,02
-0,11	0,91	-0,22	-17,95	-8,24	-6,63
0,00	1,01	0,00	-6,44	-18,07	-0,28
-0,07	0,90	-0,02	3,73	-9,28	3,66
-0,17	0,91	0,06	-0,32	12,43	7,28
-0,13	0,99	-0,02	29,41	-31,05	27,61
0,44	1,01	-0,36	-21,23	16,72	-13,20
0,23	0,98	-0,07	0,76	-111,80	3,73
0,04	0,76	-0,09	-5,56	-132,18	10,78

ax	ay	az	gx	gy	gz
0,09	0,95	-0,09	-19,38	-91,48	5,01
0,07	1,15	0,02	-34,79	-82,67	-8,93
-0,04	0,95	-0,07	13,89	-103,68	6,36
-0,08	0,92	-0,06	26,25	-58,64	10,89
-0,04	0,94	-0,06	-0,81	-2,74	10,89
0,19	0,91	-0,24	-13,82	-20,18	-0,40
0,12	0,82	-0,10	-4,04	-63,60	1,30
0,12	0,88	0,02	-8,62	-26,30	-8,98
0,01	1,21	0,00	-29,10	-27,66	-28,11
-0,05	1,11	-0,21	8,92	-23,34	7,21
-0,02	0,89	0,03	21,12	-3,35	5,63
0,00	0,96	0,04	-4,07	27,58	10,82
0,00	1,09	-0,07	-12,90	55,70	2,85
0,02	0,85	-0,05	-6,07	48,89	-11,65
0,09	0,85	0,02	-0,32	5,77	-10,38
0,17	1,02	0,11	-15,58	-10,40	-11,50
-0,11	0,76	-0,19	2,98	20,89	5,60
-0,05	0,83	-0,03	23,83	-5,56	17,97
0,02	0,92	-0,02	0,50	35,54	7,98
-0,08	0,94	0,01	-5,81	65,56	3,11
0,43	1,06	-0,19	7,25	2,00	-5,26
0,11	0,82	-0,10	-4,70	24,92	3,60
0,13	0,93	0,13	4,34	4,26	-8,60
0,13	1,00	0,03	2,05	-10,01	-11,46
-0,13	0,85	-0,17	11,17	-14,63	13,40
0,01	0,86	-0,04	22,85	-21,82	14,11

ax	ay	az	gx	gy	gz
0,01	0,93	-0,01	-1,64	-5,55	9,32
0,15	1,31	0,20	-18,39	-12,83	0,72
0,10	1,10	-0,04	-0,17	-52,97	-3,60
0,09	0,93	0,02	-5,20	-76,66	5,64
0,07	0,96	0,06	-11,89	-119,84	-3,28
-0,25	0,72	0,00	7,62	-99,20	-5,85
-0,11	0,87	-0,22	14,64	-170,65	13,11
-0,08	0,94	-0,16	-6,56	-137,84	8,08
-0,07	0,95	-0,19	-0,92	-132,03	8,91
0,14	1,00	-0,27	-1,57	-182,95	7,79
0,13	0,92	-0,04	-2,97	-98,69	15,52
0,16	0,94	-0,02	-1,16	-68,24	6,89
0,13	1,05	-0,07	-21,08	-22,44	-7,16
0,03	0,88	-0,18	3,74	2,85	7,86
0,05	0,85	-0,08	5,92	25,96	12,05
0,17	0,98	-0,02	2,89	56,44	3,65
0,21	1,20	0,04	-16,60	33,87	-5,76
0,17	1,05	0,03	9,60	27,50	-12,74
0,16	0,88	0,09	12,30	2,14	-8,54
0,21	1,15	-0,06	-23,69	-30,30	-15,60
0,07	1,05	-0,11	14,84	-93,77	6,01
-0,02	0,90	-0,07	7,02	-34,28	9,73
-0,04	0,89	-0,01	4,11	9,00	12,58
-0,05	1,30	-0,15	-14,32	32,15	8,66
0,30	0,96	-0,18	-2,82	108,94	-13,37
0,15	0,84	0,01	-7,88	-16,40	-3,72

ax	ay	az	gx	gy	gz
0,16	0,90	0,07	-0,21	-22,44	-0,21
0,13	0,99	0,05	-1,36	-27,34	-12,60
-0,05	0,78	-0,10	-9,05	-20,02	2,41
0,08	0,82	-0,03	10,71	-28,84	12,32
0,09	0,90	0,03	4,11	-8,85	9,80
-0,02	1,13	0,06	-6,99	1,76	-0,53
0,39	0,82	-0,22	6,63	-39,15	-9,45
0,22	0,80	0,07	-9,40	-60,85	-4,85
0,10	0,92	0,12	-7,67	-38,05	-9,65
0,04	1,10	0,13	-21,11	-30,79	-10,43
-0,01	1,14	-0,24	11,06	-77,84	15,02
0,00	0,86	-0,05	8,12	-0,57	9,23
0,06	0,96	-0,07	-6,55	27,78	0,21
0,16	1,01	-0,08	3,38	63,97	-13,44
0,17	0,93	-0,09	18,50	11,12	-8,93
0,16	0,93	-0,13	4,87	67,31	-6,63
-0,11	0,93	-0,09	-0,01	53,60	-20,50
0,06	0,97	-0,13	-0,30	122,39	4,74
0,07	0,91	-0,03	10,19	151,23	-3,53
0,07	0,97	0,00	20,17	155,73	-2,17
0,07	0,90	-0,03	17,15	152,91	-7,73
0,30	0,91	-0,12	7,86	144,92	-9,29
0,23	0,89	-0,12	27,09	149,57	-7,25
0,19	0,94	0,03	12,99	86,56	-6,35
0,11	1,05	-0,05	-16,66	43,35	-26,39
0,04	1,05	-0,22	-4,41	-0,47	-1,11

ax	ay	az	gx	gy	gz
-0,01	0,90	-0,06	4,60	-27,15	12,71
0,02	0,92	-0,05	-9,07	-35,22	6,57
0,04	1,26	-0,09	-14,60	-24,03	-4,34
0,19	0,99	-0,06	17,08	7,20	5,96
0,17	0,84	-0,02	5,73	24,13	-0,56
0,14	0,89	0,06	-33,67	27,41	-15,85
0,11	0,93	-0,19	-8,94	-4,12	-0,12
0,04	0,86	-0,13	15,58	70,34	9,85
0,01	1,02	-0,10	-6,57	136,76	1,87
0,10	0,89	-0,11	22,34	122,96	3,31
0,17	0,96	-0,09	13,92	97,15	-3,53
0,10	0,98	0,06	7,89	93,12	-12,21
0,19	0,94	0,03	4,23	57,77	0,81
0,03	0,95	-0,10	-12,53	37,40	3,15
0,02	0,88	-0,05	4,95	-5,92	-2,97
0,05	0,96	-0,06	4,27	-21,83	-0,15
0,00	0,95	-0,04	-13,76	-22,02	6,44
0,02	1,14	0,04	-27,34	-26,32	14,86
0,21	1,00	-0,13	-14,08	-70,92	-10,66
0,06	0,88	-0,01	-15,82	-101,52	-2,15
0,11	0,95	0,04	-11,34	-124,45	-6,49
0,11	0,96	0,11	-20,47	-116,73	-12,79
-0,02	0,95	0,02	-8,17	-96,94	-11,57
0,01	0,97	0,06	16,31	-72,49	-1,17
-0,01	0,94	-0,08	10,14	-41,68	-7,73
0,07	0,91	-0,09	2,16	1,27	3,10

ax	ay	az	gx	gy	gz
0,12	1,00	0,00	-0,85	-7,61	-4,53
0,08	0,95	-0,01	-5,11	-13,17	-2,60
0,12	0,97	0,00	-3,29	-8,95	-5,11
0,01	0,97	0,00	1,83	-8,85	-4,25
0,00	0,94	0,00	0,77	-6,23	-2,39
-0,01	0,94	0,01	0,18	-1,77	-3,98
-0,03	0,95	0,00	2,38	3,77	-1,95
-0,03	0,96	-0,01	1,53	-0,02	-2,33
-0,02	0,96	-0,01	-1,88	-0,31	-1,31
-0,01	0,95	-0,01	0,04	0,28	0,27
0,00	0,95	0,02	0,66	-4,57	0,34
0,00	0,96	-0,03	1,06	-3,34	0,82
-0,01	0,95	-0,01	-0,01	-1,11	-0,34
0,00	0,95	-0,02	-0,43	-3,75	-1,57
0,00	0,95	-0,01	-1,44	-1,23	-1,34
-0,01	0,95	0,00	-2,65	0,02	-1,08
-0,01	0,95	0,03	-2,66	1,28	-0,15
0,01	0,96	-0,03	-2,46	-1,22	-0,08
0,01	0,96	0,03	-2,11	0,29	0,16
-0,01	0,95	0,02	-0,23	0,97	-0,71
-0,04	0,95	0,03	-0,08	-0,60	-0,43

**Tabel 6.6 Contoh data uji coba aktivitas berjalan pada sensor
2**

ax	ay	az	gx	gy	gz
0,09	0,92	0,18	-0,21	0,73	0,37

ax	ay	az	gx	gy	gz
0,10	0,93	0,17	0,12	0,59	0,41
0,09	0,92	0,15	5,47	-4,64	1,76
0,11	0,93	0,15	2,92	2,76	2,64
0,09	0,93	0,14	-0,61	5,79	1,74
0,11	0,93	0,15	-2,43	3,38	0,96
0,11	0,92	0,17	-1,81	5,37	3,23
0,10	0,97	0,29	-56,47	-27,94	-10,31
0,14	0,83	0,41	-59,83	-8,76	8,30
0,15	0,67	0,49	3,66	-2,63	1,67
0,11	0,81	0,59	64,21	-2,60	-23,91
0,14	0,94	0,28	28,11	7,98	-28,73
0,01	0,96	0,12	36,15	-4,91	-6,59
0,01	0,92	0,19	13,79	35,76	2,99
0,02	0,90	0,17	30,53	52,31	9,58
-0,01	1,14	0,11	-11,72	66,37	-7,18
0,31	0,98	0,31	-120,14	20,15	21,29
0,11	0,74	0,36	-45,17	-27,26	-4,94
0,02	0,81	0,68	37,80	-19,50	-18,41
0,11	1,24	0,26	6,74	-20,31	-7,21
0,21	0,94	0,14	33,89	-46,72	-2,66
-0,11	0,90	0,23	38,90	-17,91	10,04
0,00	0,91	0,13	-53,50	2,89	-24,36
0,05	0,87	0,24	-8,48	-29,24	-12,44
-0,07	1,43	0,36	-5,70	-43,36	2,55
-0,10	0,95	0,19	12,92	69,89	10,24
-0,06	0,93	0,19	30,55	-3,02	21,06

ax	ay	az	gx	gy	gz
-0,04	0,90	0,01	21,21	20,44	29,50
0,06	1,03	0,08	21,64	-51,24	39,36
0,44	0,90	0,25	-139,92	41,79	-12,56
0,39	0,99	0,09	-71,84	-112,11	-81,97
0,31	0,64	0,25	-45,17	-112,84	-42,68
0,08	0,80	0,60	-9,90	-94,69	-48,91
0,17	0,94	0,47	36,72	-70,50	-52,47
0,32	0,87	0,12	35,82	20,25	-6,47
0,06	0,88	0,18	28,74	-10,20	-3,45
-0,07	0,98	0,17	27,63	15,89	7,64
0,15	1,00	-0,11	-152,96	-44,95	-32,00
0,26	0,72	0,24	-72,36	-142,02	-21,62
-0,03	0,64	0,73	56,11	48,48	2,64
0,03	1,35	-0,34	34,95	5,53	13,79
0,34	0,91	0,25	51,28	-55,10	-8,79
-0,06	0,88	0,20	30,76	35,45	4,31
-0,10	0,92	0,06	37,63	53,58	5,69
-0,17	1,14	0,31	-96,63	31,13	-42,30
0,06	0,73	0,19	-99,15	-142,82	-9,00
0,09	0,52	0,40	41,46	44,16	4,15
0,36	0,69	0,91	-9,69	-36,31	-25,50
-0,89	1,15	-0,30	79,54	17,18	-0,96
0,01	0,86	0,19	45,92	77,54	8,27
0,04	0,91	0,07	41,96	82,82	13,40
-0,25	0,99	0,12	17,87	97,18	32,66
0,48	1,28	0,42	-173,32	-7,82	-13,56

ax	ay	az	gx	gy	gz
0,20	0,65	0,27	-77,08	-4,63	-14,60
0,12	0,65	0,40	33,18	15,97	7,26
0,39	0,99	0,81	-16,09	-101,32	-13,84
-0,40	1,06	-0,03	52,70	-54,11	-1,51
0,03	0,86	0,20	46,31	9,06	10,98
-0,06	0,90	0,16	35,93	45,21	11,29
-0,12	1,18	0,37	-22,28	-2,98	-6,53
-0,28	1,18	0,06	-95,92	22,59	-34,42
0,11	0,69	0,22	-21,61	-102,21	-37,76
0,21	1,04	0,49	18,92	-160,74	-40,29
0,49	0,66	0,31	11,50	-104,55	12,89
-0,17	0,94	0,16	-3,51	-101,14	-33,04
0,05	0,96	0,10	4,50	-124,71	-18,98
0,12	0,94	0,19	-11,82	-109,84	-0,78
-0,01	0,89	0,41	-52,79	-138,62	-31,82
0,38	0,84	0,19	-79,69	-171,63	-57,97
0,12	0,73	0,45	1,30	-59,18	-21,64
0,30	0,90	0,52	12,61	-57,82	-18,82
0,08	0,91	0,00	43,06	-42,86	-10,73
0,03	0,90	0,18	32,42	67,55	4,15
-0,06	0,93	0,15	26,98	83,19	11,61
-0,14	1,12	0,35	-28,78	92,39	-6,08
-0,25	1,06	0,17	-82,31	-4,52	5,49
0,12	0,47	0,39	6,61	-26,92	-5,27
0,42	0,99	0,50	15,10	-13,39	-6,37
0,45	0,83	0,29	46,81	-49,23	-0,62

ax	ay	az	gx	gy	gz
0,08	0,90	0,16	48,56	6,10	8,51
-0,01	0,90	0,15	32,73	37,92	12,21
-0,31	1,40	0,29	-11,05	28,39	13,31
0,56	1,03	0,26	-138,41	-11,63	-47,12
0,11	0,73	0,19	-88,76	-54,56	-14,73
0,16	0,58	0,40	26,94	-34,98	-31,08
0,30	0,99	0,92	28,82	-20,57	1,25
0,09	0,56	-0,23	40,58	13,26	-22,46
0,11	0,83	0,19	45,60	5,80	-1,19
0,11	0,89	0,07	53,63	59,03	9,52
-0,26	1,08	0,39	14,08	34,11	1,98
0,43	1,27	0,34	-174,66	-18,53	-15,07
0,15	0,73	0,26	-80,18	-127,92	-16,82
0,18	0,54	0,39	28,82	-21,07	-8,68
0,40	0,60	0,54	7,15	-61,24	-11,56
0,26	0,88	0,20	7,21	-133,20	-41,39
0,06	0,89	0,32	52,25	36,90	15,61
-0,24	0,97	0,27	18,21	55,44	13,98
0,08	0,96	0,22	-82,96	28,62	-33,98
0,10	0,81	0,16	-29,18	-31,38	14,00
-0,11	0,90	0,40	57,51	104,07	16,51
0,84	0,79	0,41	-8,84	39,64	37,28
-0,47	1,06	-0,12	1,76	95,00	24,91
-0,06	0,93	0,13	8,20	145,42	23,89
-0,12	0,93	0,17	-1,72	138,71	18,54
0,04	0,89	0,24	8,37	148,23	17,57

ax	ay	az	gx	gy	gz
0,22	0,97	-0,01	-54,54	205,41	27,74
-0,07	0,91	0,33	-33,07	113,63	65,23
-0,05	0,76	0,38	8,74	85,14	7,54
0,37	1,21	0,35	10,72	5,11	17,08
0,05	1,00	0,31	35,17	-5,78	5,43
0,05	0,88	0,26	34,56	-0,15	6,83
-0,03	0,89	0,12	44,82	28,83	21,15
0,09	1,33	0,10	-35,79	19,86	-17,69
-0,22	0,98	0,05	-89,78	-7,68	-26,66
0,07	0,57	0,23	2,29	4,76	-3,54
-0,06	0,82	0,57	27,41	72,31	-10,37
-0,14	0,95	-0,18	44,58	51,92	8,76
0,16	0,92	0,13	34,78	89,40	19,74
-0,06	0,96	0,12	-0,55	122,71	25,84
0,00	0,95	0,15	-24,07	45,07	15,22
0,19	0,91	0,36	-8,98	139,94	57,77
0,25	0,79	0,38	4,86	50,71	59,44
0,23	0,72	0,50	42,23	83,49	15,78
-0,16	0,89	0,04	24,92	27,68	-3,89
0,02	0,89	0,22	18,65	56,44	6,39
-0,03	0,92	0,25	19,03	10,67	-0,10
0,04	0,91	0,14	40,47	6,33	11,35
-0,31	1,22	0,17	-24,04	-1,25	3,21
-0,35	1,18	0,14	-127,37	13,13	-27,85
-0,04	0,74	0,24	-62,85	-147,72	-80,02
0,24	0,70	0,47	13,39	-99,37	-55,00

ax	ay	az	gx	gy	gz
0,15	0,78	0,37	22,63	-64,66	-57,86
-0,36	0,94	0,05	41,00	-62,18	-21,24
0,07	0,95	0,12	22,98	-24,27	-17,89
0,07	0,92	0,26	12,13	-3,40	0,41
0,05	1,01	0,18	-52,40	-31,87	-15,93
0,07	0,83	0,28	-29,04	-45,95	2,52
0,22	0,89	0,23	52,95	-14,38	3,92
-0,17	0,88	0,01	26,90	42,98	18,33
0,17	0,90	0,25	8,57	36,92	0,47
0,17	0,92	0,17	6,35	-1,48	-3,67
0,13	0,93	0,15	5,51	-3,87	-3,37
0,09	0,93	0,13	5,28	-0,24	-1,26
0,10	0,92	0,14	1,13	-2,18	-2,13
0,10	0,93	0,10	2,28	-3,50	-0,67
0,11	0,93	0,11	1,60	0,15	-0,04
0,11	0,93	0,10	0,11	-2,05	-0,50
0,10	0,94	0,10	0,44	-2,15	-0,04
0,12	0,93	0,10	0,10	-1,59	-0,11
0,11	0,93	0,10	-0,47	-0,25	-0,18
0,11	0,94	0,09	-0,73	-1,31	-0,36
0,12	0,93	0,09	0,08	1,98	-0,21
0,09	0,93	0,09	-0,65	3,47	-0,22
0,08	0,93	0,11	-1,21	1,27	-0,13
0,08	0,94	0,10	-1,91	-1,11	-0,63
0,10	0,94	0,12	0,47	-0,19	-0,35
0,09	0,93	0,12	-1,63	0,89	-0,20

Tabel 6.7 Contoh data uji coba gerakan transisi pada sensor 2

ax	ay	az	gx	gy	gz
0,02	0,93	0,10	-1,08	-3,40	-0,44
0,01	0,94	0,11	-0,92	-1,95	-0,31
0,03	0,94	0,09	-1,48	-4,92	-1,04
0,01	0,95	0,12	1,23	-0,50	1,57
0,00	0,93	0,11	0,50	0,33	0,69
0,05	0,94	0,06	-1,03	2,82	-0,27
0,02	0,94	0,10	-1,40	1,46	-0,41
0,05	0,94	0,10	-0,82	-1,83	0,05
0,03	0,94	0,11	-1,03	-0,14	-0,08
0,02	0,94	0,11	-0,93	-0,97	-0,36
0,02	0,93	0,10	-1,11	-15,47	-2,14
0,02	0,94	0,13	-5,16	4,63	0,45
0,01	0,95	0,08	-7,98	-5,51	5,89
0,10	0,93	0,04	-18,61	-5,18	-1,07
0,18	0,88	0,25	-36,71	-34,53	-3,21
0,04	0,84	0,33	-27,90	-15,74	-0,85
0,18	0,72	0,21	25,62	4,56	-8,01
-0,95	0,73	0,32	-59,37	146,05	8,15
0,44	0,80	0,61	-83,81	-20,84	-13,21
0,21	0,58	0,83	-71,80	31,69	14,31
0,19	0,49	1,03	-65,83	-11,73	-1,15
0,18	0,25	1,03	-53,89	-3,67	25,08
0,20	0,07	1,06	-60,05	0,51	18,94
0,20	-0,12	1,02	-63,56	-1,18	25,69
0,27	-0,41	1,18	-46,96	-33,79	22,54

ax	ay	az	gx	gy	gz
0,36	-0,47	0,88	-14,35	-2,09	12,74
0,19	-0,49	0,99	-11,82	11,13	-3,57
0,14	-0,48	0,96	-3,85	19,23	-2,83
0,20	-0,51	0,99	-22,08	-21,72	4,15
0,24	-0,54	0,91	-20,55	-9,59	-13,07
0,30	-0,55	0,88	-6,34	-9,09	-9,74
0,28	-0,55	0,85	-3,35	-2,66	-1,40
0,28	-0,57	0,88	-0,20	0,80	-2,16
0,30	-0,58	0,85	-1,15	2,78	-0,07
0,28	-0,59	0,87	0,53	2,65	3,59
0,28	-0,59	0,87	-2,14	2,81	-0,28
0,27	-0,59	0,87	1,21	-0,18	1,82
0,27	-0,57	0,87	-1,09	-0,04	-1,18
0,29	-0,56	0,87	-1,75	-0,24	-2,31
0,28	-0,58	0,85	-1,43	0,07	-0,09
0,28	-0,57	0,87	-0,09	0,03	0,57
0,29	-0,58	0,87	-0,38	0,16	0,18
0,29	-0,58	0,86	-1,32	1,06	0,50
0,28	-0,56	0,86	-2,10	1,18	0,35
0,27	-0,57	0,87	-2,68	0,46	-0,74
0,28	-0,57	0,87	3,83	-1,18	-1,40
0,28	-0,59	0,86	1,98	-0,62	-2,95
0,28	-0,59	0,87	6,59	-0,98	-5,34
0,31	-0,56	0,88	4,29	-1,98	-4,53
0,31	-0,53	0,87	6,02	-1,09	-5,56
0,38	-0,52	0,93	7,21	3,49	-3,34

ax	ay	az	gx	gy	gz
0,37	-0,50	1,18	21,84	50,37	-17,45
0,40	-0,27	1,09	76,83	4,87	-27,96
0,25	-0,08	1,03	92,81	10,14	-21,79
0,23	0,20	0,93	90,24	18,95	-17,73
0,16	0,45	0,77	105,40	-11,52	-13,58
0,21	0,63	0,54	92,35	10,58	-20,80
0,23	0,74	0,38	75,94	44,80	-7,53
0,07	0,89	0,06	38,51	67,78	16,31
0,05	0,91	0,16	7,37	25,91	3,16
0,05	0,93	0,16	-11,40	-5,15	3,36
0,08	0,94	0,12	6,61	-15,63	-0,62
0,08	0,94	0,15	5,92	-10,40	-0,40
0,05	0,93	0,12	-0,18	0,27	1,28
0,07	0,94	0,12	0,41	2,34	1,49
0,08	0,94	0,12	0,05	4,52	1,02
0,07	0,93	0,13	-1,08	1,65	0,31
0,08	0,94	0,15	0,52	1,60	1,01
0,07	0,92	0,13	-0,82	0,40	0,53
0,07	0,93	0,12	-2,13	-3,45	-0,34
0,08	0,94	0,12	-2,32	0,30	0,04
0,07	0,93	0,17	-0,20	3,24	0,74
0,07	0,93	0,16	-0,48	-0,63	-0,26

Tabel 6.8 Contoh data uji coba gerakan merebahkan badan pada sensor 1

ax	ay	az	gx	gy	gz
-0,04	1,05	0,01	1,50	9,79	4,05
0,02	0,98	0,03	5,70	6,87	1,29
0,11	0,92	-0,09	0,92	35,21	-3,60
0,05	0,96	-0,06	9,73	9,18	1,14
0,08	0,95	-0,07	-3,17	1,66	1,69
0,11	0,97	-0,05	-3,54	-5,00	-0,28
0,06	0,94	-0,07	3,51	-6,08	0,81
0,08	0,96	-0,09	7,26	-2,19	-0,48
0,02	0,91	-0,11	22,23	-0,91	11,07
0,10	0,83	-0,21	33,76	13,76	8,00
0,15	0,79	-0,32	27,75	-25,05	6,16
0,15	0,73	-0,33	18,95	-35,91	18,15
0,15	0,74	-0,38	0,13	10,82	6,50
0,11	0,95	-0,40	-12,53	0,48	-1,56
0,23	0,96	-0,36	-47,48	3,40	-8,46
0,14	1,36	-0,17	-85,32	-0,13	-21,13
0,13	0,73	0,17	-87,50	-16,48	-7,24
0,06	0,86	0,35	-107,91	5,32	-8,40
0,01	0,53	0,61	-114,89	25,73	-14,98
-0,15	0,26	1,14	-105,82	31,37	-27,18
-0,13	0,18	1,02	-21,78	6,63	-19,62
-0,31	0,00	0,93	-14,84	25,99	-7,60
-0,29	0,11	1,02	-32,77	-0,19	-4,27
-0,31	0,01	1,01	21,17	-16,72	2,54

ax	ay	az	gx	gy	gz
-0,23	0,02	0,99	1,80	0,24	1,06
-0,21	0,05	0,95	2,65	-2,15	0,82
-0,24	0,03	0,93	1,31	-0,55	-0,42
-0,21	0,06	0,93	-1,83	-3,20	0,10
-0,21	0,04	0,95	-2,00	0,98	-0,14
-0,21	0,06	0,95	-0,20	0,15	-0,31
-0,20	0,04	1,00	-4,98	0,02	0,41
-0,21	0,05	0,94	-0,15	-1,63	0,04
-0,20	0,08	0,97	-1,96	1,20	-1,32
-0,21	0,05	0,91	0,95	26,61	4,57
-0,31	0,08	0,94	16,89	-5,63	5,21
-0,20	0,10	0,88	1,62	0,23	4,27
-0,19	0,11	0,94	-11,92	-8,27	-6,97
-0,26	0,08	0,99	-10,76	-1,40	4,21
-0,22	0,08	0,93	-0,04	-8,82	-1,08
-0,25	0,11	0,93	4,78	-0,48	1,45
-0,22	0,12	0,93	4,89	-1,27	-1,21
-0,22	0,11	0,94	-1,30	-1,44	0,88
-0,25	0,12	0,94	2,24	-2,60	-0,45
-0,22	0,11	0,94	-1,64	-0,88	0,27
-0,19	0,10	0,97	-3,76	5,27	-0,17
-0,29	0,10	0,99	-4,84	9,98	0,79
-0,28	0,10	0,98	6,60	5,01	-0,56
-0,28	0,13	0,94	1,75	0,48	-1,79

Tabel 6.9 Contoh data uji coba gerakan merebahkan badan pada sensor 2

ax	ay	az	gx	gy	gz
-0,01	0,95	0,07	-2,68	7,02	4,23
0,01	0,94	0,14	-4,05	15,66	6,60
0,06	0,93	0,14	-1,93	15,49	8,63
0,00	0,92	0,22	1,49	20,10	8,55
0,07	0,94	0,20	-2,33	2,33	2,40
0,08	0,93	0,16	0,07	-4,64	0,97
0,11	0,93	0,16	-0,34	-3,24	0,20
0,08	0,95	0,15	-5,56	-4,64	-0,12
0,01	0,91	0,14	-22,82	10,04	3,58
0,00	0,91	0,23	-47,37	-37,55	-4,56
0,25	0,75	0,31	-63,20	-35,55	9,14
0,49	0,48	0,66	-110,80	12,92	20,79
0,05	0,50	0,79	-119,87	5,67	-2,66
0,19	0,24	1,04	-137,73	-40,84	21,46
0,19	-0,31	0,95	-126,66	-15,24	41,16
0,56	-0,80	1,23	-99,52	-8,45	49,59
-0,01	-1,00	0,70	-11,20	36,37	2,41
-0,05	-0,64	0,89	-61,99	18,73	18,02
-0,21	-1,05	0,49	-101,56	36,81	10,56
-0,20	-1,12	0,14	-98,14	39,03	7,13
-0,26	-0,87	-0,25	-39,32	-15,44	19,76
-0,37	-1,15	-0,06	-21,32	-46,94	12,73
-0,30	-1,02	-0,16	-23,36	-5,73	-3,92
-0,29	-0,96	-0,28	-7,73	7,56	-4,15

ax	ay	az	gx	gy	gz
-0,30	-0,98	-0,25	0,92	2,21	-4,41
-0,26	-0,98	-0,22	1,14	-1,06	-5,09
-0,26	-0,98	-0,22	0,04	0,92	-2,05
-0,26	-1,00	-0,23	-4,50	0,31	-0,90
-0,27	-0,98	-0,24	-0,91	-0,74	1,18
-0,27	-0,98	-0,25	0,79	-0,27	-0,03
-0,26	-0,98	-0,23	-0,94	0,11	-0,10
-0,27	-0,97	-0,24	0,34	-0,45	0,21
-0,27	-0,98	-0,23	-0,66	-5,99	-3,56
-0,26	-0,99	-0,22	-3,10	-2,91	-10,79
-0,21	-0,99	-0,25	-4,12	0,21	-8,89
-0,20	-0,98	-0,25	2,31	-4,02	-5,18
-0,18	-1,00	-0,23	1,01	-1,16	1,27
-0,19	-1,00	-0,24	-2,00	-2,56	3,89
-0,21	-1,00	-0,25	-1,10	-1,89	0,75
-0,20	-1,01	-0,24	2,28	0,98	-1,15
-0,20	-1,00	-0,24	-1,69	-1,21	-0,23
-0,19	-1,01	-0,26	-4,18	0,36	-3,53
-0,18	-1,01	-0,27	-0,60	0,07	-2,90
-0,18	-0,99	-0,26	-2,32	-0,53	-1,68
-0,18	-0,99	-0,27	-2,88	-0,64	-0,39
-0,19	-1,00	-0,27	1,79	-0,73	2,41
-0,18	-0,99	-0,24	-0,05	-1,11	0,37
-0,18	-0,99	-0,26	0,61	0,79	1,14

Tabel 6.10 Contoh data uji coba gerakan jatuh pada sensor 1

ax	ay	az	gx	gy	gz
-0,19	0,89	0,26	0,80	-4,84	-3,37
-0,19	0,90	0,28	-0,50	1,15	-2,56
-0,20	0,89	0,28	2,36	1,35	-0,94
-0,19	0,89	0,22	3,40	5,98	2,73
-0,19	0,90	0,23	6,19	1,93	2,92
-0,19	0,92	0,22	3,31	2,38	2,79
-0,18	0,91	0,21	4,44	-0,23	1,18
-0,17	0,90	0,20	6,38	-1,48	0,26
-0,18	0,91	0,16	5,44	7,47	0,91
-0,20	0,92	0,15	5,44	2,66	2,15
-0,16	0,93	0,08	4,60	1,32	3,34
-0,15	0,92	0,08	-3,22	4,60	1,50
-0,13	0,88	0,06	0,92	-8,08	2,29
-0,10	0,60	0,07	9,19	-17,62	-2,51
0,04	0,47	-0,04	2,96	-14,10	-12,94
0,03	0,47	0,09	-19,55	11,79	6,31
-0,11	1,19	0,00	-57,66	10,47	-29,85
-0,83	1,54	0,46	-46,85	-54,47	-18,38
0,05	-0,72	0,56	-250,14	-83,89	-35,41
-0,06	-0,17	0,88	53,47	-44,70	5,40
0,19	-0,34	1,07	-24,28	4,94	-14,25
0,02	0,11	1,08	27,95	8,07	-1,69
-0,13	0,13	0,99	24,68	5,84	1,28
-0,13	0,12	0,98	-12,18	0,78	-6,58
-0,27	0,37	1,08	-21,31	-1,41	-1,11

ax	ay	az	gx	gy	gz
-0,17	0,12	0,96	1,80	25,85	-1,85
-0,13	-0,03	0,97	-2,44	11,64	8,28
-0,22	0,04	1,00	-4,02	7,93	-1,24
0,04	0,26	1,01	-0,44	-15,02	-9,34
-0,13	-0,06	1,03	8,19	24,11	-7,64
-0,22	0,05	0,97	7,64	-7,62	-6,49
-0,21	0,09	0,98	1,47	-2,96	0,03
-0,15	0,11	0,95	5,73	-6,84	4,48
-0,15	0,15	0,95	2,20	1,33	-1,16
-0,06	0,10	0,95	-0,03	-2,75	-2,47
-0,14	0,08	0,94	-6,96	-0,56	-3,85
-0,18	0,09	0,95	-3,53	1,96	-0,74
-0,15	0,09	0,95	-0,56	-3,05	-2,95
-0,15	0,09	0,97	-3,91	-2,37	1,38
-0,13	0,07	0,97	-3,72	-1,31	0,10
-0,14	0,08	0,96	-2,44	0,44	-0,08
-0,15	0,07	1,02	-0,24	-1,04	-0,19
-0,16	0,10	0,97	-2,17	-0,08	1,39

BIODATA PENULIS



Buthoro Kunto Raharjo, lahir pada tanggal 23 Maret 1996 di Sukoharjo, Jawa Tengah. Penulis merupakan seorang mahasiswa yang sedang menempuh studi di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Penulis memiliki hobi di bidang editing, olahraga dan bermain game. Penulis memiliki beberapa pengalaman dalam organisasi seperti menjabat sebagai Staff Departemen Dalam Negeri Himpunan

Mahasiswa Teknik Computer-Informatika (HMTTC) ITS periode 2015-2016 dan Wakil Ketua Ikatan Mahasiswa Sukoharjo di Surabaya (IKEMAS) periode 2016-2017. Penulis juga memiliki beberapa pengalaman dalam hal event organizing seperti menjadi staff National Programing Competition Big Event Schematics periode 2014-2015, staff ahli National Programing Competition Big Event Schematics periode 2015-2016, dan lain-lain. Penulis dapat dihubungi melalui email: buthororaharjo@gmail.com.